
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS

```
PPPPPPPP      AAAAAA      GGGGGGGG  EEEEEEEEEE  FFFFFFFFFF    AAAAAA    UU      UU  LL          TTTTTTTTTT
PPPPPPPP      AAAAAA      GGGGGGGG  EEEEEEEEEE  FFFFFFFFFF    AAAAAA    UU      UU  LL          TTTTTTTTTT
PP      PP   AA      AA  GG        EE          FF          AA      AA  UU      UU  LL              TT
PP      PP   AA      AA  GG        EE          FF          AA      AA  UU      UU  LL              TT
PP      PP   AA      AA  GG        EE          FF          AA      AA  UU      UU  LL              TT
PP      PP   AA      AA  GG        EE          FF          AA      AA  UU      UU  LL              TT
PPPPPPPP     AA      AA  GG        EEEEEEEE    FFFFFFFF    AA      AA  UU      UU  LL              TT
PPPPPPPP     AA      AA  GG        EEEEEEEE    FFFFFFFF    AA      AA  UU      UU  LL              TT
PP           AAAAAAAAAA  GG  GGGGG EE          FF          AAAAAAAAAA  UU      UU  LL              TT
PP           AAAAAAAAAA  GG  GGGGG EE          FF          AAAAAAAAAA  UU      UU  LL              TT
PP           AA      AA  GG    GG  EE          FF          AA      AA  UU      UU  LL              TT          ....
PP           AA      AA  GG    GG  EE          FF          AA      AA  UU      UU  LL              TT          ....
PP           AA      AA    GGGGGG  EEEEEEEEEE  FF          AA      AA  UUUUUUUUUU  LLLLLLLLLL      TT          ....
PP           AA      AA    GGGGGG  EEEEEEEEEE  FF          AA      AA  UUUUUUUUUU  LLLLLLLLLL      TT          ....

LL             IIIIII      SSSSSSSS
LL             IIIIII      SSSSSSSS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  PAGEFAULT - TRANSLATION NOT VALID EXCEPTION HANDLER
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26
0000    27 ;++
0000    28 ; FACILITY:     EXECUTIVE, TRANSLATION NOT VALID EXCEPTION HANDLER
0000    29 ;
0000    30 ; ABSTRACT:
0000    31 ;
0000    32 ; ENVIRONMENT:
0000    33 ;
0000    34 ;--
0000    35 ;
0000    36          .SBTTL  HISTORY                 ; DETAILED
0000    37 ;
0000    38 ; AUTHOR: PETER H. LIPMAN       , CREATION DATE: 14-SEP-76
0000    39 ;
0000    40 ; MODIFIED BY:
0000    41 ;
0000    42 ;       V03-008 WMC0002         Wayne Cardoza           24-Jul-1984
0000    43 ;               Comparison on modified page list waitlimit should be CMPL.
0000    44 ;
0000    45 ;       V03-007 WMC0001         Wayne Cardoza           05-MAY-1983
0000    46 ;               Fix trigger for dead page table scan to account for
0000    47 ;               locked page tables in dynamic part of working set.
0000    48 ;
0000    49 ;       V03-006 TCM0001         Trudy C. Matthews       31-Mar-1983
0000    50 ;               Change references to working set fields in PHD so that
0000    51 ;               they are used as unsigned words.
0000    52 ;
0000    53 ;       V03-005 KDM54836        Kathleen D. Morse       16-Mar-1983
0000    54 ;               Add deadlock detection:  do not allow process holding
0000    55 ;               a mutex to wait for swap file space.
0000    56 ;
0000    57 ;       V03-004 KDM0002         Kathleen D. Morse       28-Jun-1982
```

```
0000    58 ;                Added $PRTDEF.
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;
                        $CADEF                          ;CONDITIONAL ASSEMBLY DEFINITIONS
                        $IPLDEF                         ;PROCESSOR PRIORITY LEVEL DEFINITIONS
                        $IRPDEF                         ;I/O REQUEST PACKET DEFINITIONS
                        $PCBDEF                         ;PROCESS CONTROL BLOCK DEFINITIONS
                        $PHDDEF                         ;PROCESS HEADER DEFINTIONS
                        $PRDEF                          ;PROCESSOR REGISTER DEFINITIONS
                        $PFLDEF                         ;PAGE FILE CONTROL BLOCK DEFINITIONS
                        $PFNDEF                         ;PFN DATA BASE DEFINITIONS
                        $PRTDEF                         ;PROTECTION FIELD DEFINITIONS
                        $PSLDEF                         ;PROCESSOR STATUS LONG WORD DEFINITIONS
                        $PTEDEF                         ;PAGE TABLE ENTRY OFFSETS
                        $RSNDEF                         ;RESOURCE NAME DEFINITIONS
                        $SECDEF                         ;SECTION TABLE DEFINITIONS
                        $SSDEF                          ;SYSTEM STATUS DEFINITIONS
                        $VADEF                          ;VIRTUAL ADDRESS VIELDS
                        $WQHDEF                         ;WAIT QUEUE HEADER DEFINITIONS
                        $WSLDEF                         ;WORKING SET LIST DEFINITIONS
                ;
                ; EXTERNAL SYMBOLS:
                ;

                ;
                ; MACROS:
                ;

                ;
                ; EQUATED SYMBOLS:
                ;

                        $OFFSET <4*6>,POSITIVE,<-
                        FLTCTL,-                        ;OFFSET TO FAULT CONTROL BITS
                        FLTVA,-                         ;OFFSET TO FAULT VIRTUAL ADDRESS
                        FLTPC,-                         ;OFFSET TO FAULT PC
                        FLTPSL-                         ;OFFSET TO FAULT PSL
                        >
0018            FLTCTL:
001C            FLTVA:
0020            FLTPC:
0024            FLTPSL:

                        $VIELD  PGF,0,<-                ;DEFINE PAGE FAULT CONTROL BITS
                                LENVIO,-                ;LENGTH VIOLATION
                                PGTBFLT,-               ;PAGE TABLE FAULT
                                WRTACC-                 ;REFERENCE WAS WRITE OR MODIFY
                        >
                ;
                ; OFFSETS INTO I/O PACKET WHILE BEING USED AS SCRATCH STORAGE FOR CLUSTERING
                ;
                        $OFFSET 0,POSITIVE,<-
                                PTEDAT,-                ;MASTER PTE CONTENTS
                                SVAPTE,-                ;MASTER PTE ADDRESS
                                SIZE_TYPE,-             ;RESERVED FOR SIZE AND TYPE
                                VA,-                    ;VIRTUAL ADDRESS
```

```
          0000                  AST,-                    ;AST INFO
          0000                  ASTPRM,-                 ; AND PARAMETER
          0000                  GPTX,-                   ;PROCESS PTE CONTENT FOR GLOBAL PAGE
          0000                  GPTX_PTE,-               ;PROCESS PTE ADR FOR GLOBAL PAGE
          0000                  <CLUSTER,1>,-            ;DESIRED CLUSTER SIZE
          0000                  <COUNT,1>,-              ;CURRENT COUNT OF PAGES
          0000                  <STATE,1>,-              ;SAVED PFN STATE BYTE
          0000                  <PRI,1>,-                ;PRIORITY OF I/O TRANSFER
          0000                  BAK,-                    ;PFN BACKING STORE ADDRESS
          0000                  INC1,-                   ;+ OR - 1
          0000                  INC4,-                   ;+ OR - 4
          0000                  INC512,-                 ;+ OR - 512
          0000                  VBN,-                    ;VIRTUAL BLOCK NUMBER
          0000                  WINDOW,-                 ;WINDOW CONTROL BLOCK ADR
          0000                  FP_SAV,-                 ;SAVED FP
          0000                  <PCB_SAV,8>,-            ;SAVED PCB, PHD ADDRESS
          0000                  PHVREFCADR,-             ;PROCESS HEADER REFERENCE COUNT ADDRESS
          0000                  <CLU_SCRATCH_SIZ,0>-     ;SIZE OF THIS SCRATCH AREA
          0000                  >
          0000      PTEDAT:
          0004      SVAPTE:
          0008      SIZE_TYPE:
          000C      VA:
          0010      AST:
          0014      ASTPRM:
          0018      GPTX:
          001C      GPTX_PTE:
          0020      CLUSTER:
          0021      COUNT:
          0022      STATE:
          0023      PRI:
          0024      BAK:
          0028      INC1:
          002C      INC4:
          0030      INC512:
          0034      VBN:
          0038      WINDOW:
          003C      FP_SAV:
          0040      PCB_SAV:
          0048      PHVREFCADR:
          004C      CLU_SCRATCH_SIZ:
          0000
          0000              ASSUME  CLU_SCRATCH_SIZ LE IRP$C_LENGTH
          0000              ASSUME  SIZE_TYPE EQ IRP$W_SIZE
          0000              ASSUME  AST EQ IRP$L_AST
          0000              ASSUME  ASTPRM EQ IRP$L_ASTPRM
          0000              ASSUME  PRI EQ IRP$B_PRI
          0000  ;
          0000  ; OWN STORAGE:
          0000  ;
      00000000              .PSECT  $$$210,LONG
          0000  ;
          0000  ; STATISTICS
          0000  ;
          0000  PMS$GL_FAULTS::
00000004  0000              .BLKL   1                        ;COUNT OF PAGE FAULTS
          0004  PMS$GL_RDFLTS::                              ;OBSOLETE NAME
```

```
                 0004   148 PMS$GL_PREADS::
00000008         0004   149         .BLKL   1                       ;PAGE READS
                 0008   150 PMS$GL_PREADIO::
0000000C         0008   151         .BLKL   1                       ;I/O REQUESTS TO READ THE PAGES
                 000C   152 PMS$GL_PWRITES::
00000010         000C   153         .BLKL   1                       ;MODIFIED PAGES WRITTEN
                 0010   154 PMS$GL_PWRITIO::
00000014         0010   155         .BLKL   1                       ;I/O REQUESTS TO WRITE THE MODIFIED PAGES
                 0014   156 PMS$GL_DZROFLTS::
00000018         0014   157         .BLKL   1                       ;DEMAND ZERO PAGE FAULTS
                 0018   158 PMS$AL_TRANSFLT::
00000038         0018   159         .BLKL   <1@PFN$S_LOC>           ;PAGE FAULTS OUT OF TRANSITION STATES
                 0038   160 PMS$GL_DPTSCN::
0000003C         0038   161         .BLKL   1                       ;DEAD PAGE TABLE SCANS
                 003C   162 PMS$GL_GVALID::
00000040         003C   163         .BLKL   1                       ;GLOBAL VALID FAULTS
```

```
                        .SBTTL  PAGE FAULT HANDLER

                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ; THIS MODULE CONTAINS THE PAGEFAULT HANDLER.  IT IS ENTERED VIA A
                ; TRANSLATION-NOT-VALID FAULT.  AT THE TIME OF A FAULT, THE KERNEL
                ; STACK CONTAINS THE FOLLOWING INFORMATION:
                ;
                ;       +-------------------------------+
                ;       !        REASON MASK            ! --> BIT 0 - ALWAYS 0 FOR
                ;       +-------------------------------+                TRANS-NOT-VALID FAULTS
                ;       !   INVALID VIRTUAL ADDRESS     !     BIT 1 - 0 INDIC VIRT ADR NOT VALID
                ;       +-------------------------------+             1 INDIC ASSOC PTE NOT VALID
                ;       !  PC OF FAULTING INSTRUCTION   !     BIT 2 - 0 INDIC READ ACCESS
                ;       +-------------------------------+             1 INDIC MODIFY/WRITE ACCESS
                ;       ! PSL OF FAULTING INSTRUCTION   !
                ;       +-------------------------------+
                ;
                ; CALLING SEQUENCE:
                ;       NONE
                ;
                ;
                ; INPUT PARAMETERS:
                ;       NONE
                ;
                ; IMPLICIT INPUTS:
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;       NONE
                ;
                ; IMPLICIT OUTPUTS:
                ;       NONE
                ;
                ; COMPLETION CODES:
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;       NONE
                ;
                ;--

                ;*****************************************************************************
                ;
                ;****************** THIS ENTIRE MODULE MUST BE RESIDENT ******************
                ;
                        .PSECT  $MMG$COD,LONG
                ;
                ;*****************************************************************************
                ;
                        .LIST   MEB

                IPLHI:  BUG_CHECK PGFIPLHI,FATAL        ;IPL TOO HIGH FOR PAGE FAULT
FEFF  0000                      .WORD   ^XFEFF
0004' 0002                      .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_PGFIPLHI!4
```

```
                     0004   221          .SBTTL  SYSTEM PAGE FAULT, ESTABLISH PAGE TYPE
                     0004   222
                     0004   223          .ENABL  LSB
                     0004   224 ;
                     0004   225 ; BAD SYSTEM PAGE - PROCESS HEADER OR PAGE TABLE PAGE FOR ANOTHER PROCESS
                     0004   226 ;
                     0004   227 ;  IF THE PROCESS HEADER HAS JUST BEEN INSWAPPED (PHD$V_NOACCVIO IS SET),
                     0004   228 ;  SIMPLY DISMISS THE FAULT. IN ALL OTHER CASES, REPORT AN ACCESS VIOLATION
                     0004   229 ;  EXCEPTION.
                     0004   230 ;
                     0004   231 ;       IF THE PROCESS WAS OUTSWAPPED WHILE ACCESSING ITS OWN HEADER,
                     0004   232 ;       DISMISSING THE EXCEPTION WILL CAUSE THE REFERENCE TO OCCUR AGAIN,
                     0004   233 ;       BUT THIS TIME TO THE CORRECT BALANCE SLOT.
                     0004   234 ;
                     0004   235 ;       IF THE PROCESS WAS MAKING AN ILLEGAL REFERENCE TO THE HEADER OF ANOTHER
                     0004   236 ;       PROCESS, DISMISSING THE EXCEPTION WILL CAUSE THE SAME ILLEGAL REFERENCE
                     0004   237 ;       TO OCCUR AGAIN, BUT NOW WITH PHD$V_ACCVIO CLEAR, CAUSING AN ACCESS
                     0004   238 ;       VIOLATION TO BE REPORTED.
                     0004   239
                     0004   240 BADSYSPAG:
 03 36 A5    03   E4     0004   241          BBSC    #PHD$V_NOACCVIO,PHD$W_FLAGS(R5),10$ ;BRANCH IF HEADER JUST INSWAPPED
          0687    31     0009   242          BRW     ACVIOLAT                    ;FAKE AN ACCESS VIOLATION
                         000C   243
          05F7    31     000C   244 10$:     BRW     PGFCOMPLETE                 ;SIMPLY DISMISS THE PAGE FAULT
                         000F   245
                         000F   246 ;
                         000F   247 ; SEE IF PAGE IS A GLOBAL PAGE TABLE PAGE, OTHERWISE ERROR
                         000F   248 ;
                         000F   249 GPGTBL:
00000000'EF    50    B1     000F   250          CMPW    R0,SGN$GL_BALSETCT          ;SYSTEM BALANCE SET SLOT?
               EC    19     0016   251          BLSS    BADSYSPAG                   ;BRANCH IF NOT
    0000'CF    52    D1     0018   252          CMPL    R2,W^MMG$GL_MAXGPTE         ;LEGAL GPTE ADDRESS
               E5    1E     001D   253          BGEQU   BADSYSPAG
          52   0A    90     001F   254          MOVB    #WSL$C_GPGTBL,R2            ;PAGE IS GLOBAL PAGE TABLE
               2E    11     0022   255          BRB     50$
                         0024   256 ;
                         0024   257 ; PAGE IS NOT SYSTEM PAGE TYPE, COULD BE GLOBAL PAGE TABLE, PROCESS PAGE TABLE
                         0024   258 ; OR PROCESS HEADER PAGE
                         0024   259 ;
                         0024   260 NOTSYSTEM:
 50   50     F7  8F    78     0024   261          ASHL    #-9,R0,R0                   ;SCALE VA DIFFERENCE TO PAGE NUMBER
      55     6C  A4    D0     0029   262          MOVL    PCB$L_PHD(R4),R5            ;ADDRESS OF PROCESS HEADER
   50     0000'CF    C6     002D   263          DIVL    W^SWP$GL_BSLOTSZ,R0         ;PROCESS HEADER INDEX
     42 A5    50    B1     0032   264          CMPW    R0,PHD$W_PHVINDEX(R5)       ;THIS PROCESS' HEADER?
               D7    12     0036   265          BNEQ    GPGTBL                      ;BRANCH IF NOT, MAYBE GLOBAL PAGE TABLE
          52   08    90     0038   266          MOVB    #WSL$C_PPGTBL,R2            ;ASSUME PROCESS PAGE TABLE
               1E    11     003B   267          BRB     70$
                         003D   268 ;
                         003D   269 ; PAGE FAULT MONITORING ENABLED FOR THIS PROCESS
                         003D   270 ;
                         003D   271 PGFMONITOR:
            FFC0'    30     003D   272          BSBW    PFM$MON                     ;CALL THE RECORDING ROUTINE
               5E    11     0040   273          BRB     PGFMONITOR1                 ;RETURN IN LINE
                         0042   274 ;
                         0042   275 ; PAGE FAULT FOR SYSTEM SPACE VIRTUAL ADDRESS
                         0042   276 ;       R2 = FAULT VA, LOW BITS CLEARED
                         0042   277 ;       R4 = PROCESS PCB ADDRESS
```

```
                                0042   278 ;
                                0042   279 $YSTEMSPACE:
      53   52   15   09   EF    0042   280          EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R3 ;PAGE NUMBER IN SYSTEM SPACE
      50   52   0000'CF   C3    0047   281          SUBL3   W^SWP$GL_BALBASE,R2,R0 ;ABOVE BASE OF BALANCE SET SLOTS?
                        D5   18 004D   282          BGEQ    NOTSYSTEM                   ;BRANCH IF NOT SYSTEM PAGE TYPE
                   52   02   90 004F   283 40$:     MOVB    #WSL$C_SYSTEM,R2            ;SYSTEM PAGE
           54   0000'CF   DE   0052   284 50$:     MOVAL   W^MMG$AL_SYSPCB,R4          ;ADDRESS OF SYSTEM PCB
              55   6C   A4   D0 0057   285          MOVL    PCB$L_PHD(R4),R5            ;ADDRESS OF SYSTEM PROCESS HEADER
         53   0000'DF43   DE   005B   286 70$:     MOVAL   @W^MMG$GL_SPTBASE[R3],R3 ;ADDRESS OF PAGE TABLE ENTRY
                        57   11 0061   287          BRB     GETPAGELOC
                                0063   288
                                0063   289 ;
                                0063   290 ; THIS IS A PROCESS PAGE TABLE FAULT
                                0063   291 ;
                                0063   292 PPGTBL:
52   53   000001FF   8F   CB   0063   293          BICL3   #^X1FF,R3,R2             ;R2 = FAULT VA
                        D5   11 006B   294          BRB     SYSTEMSPACE
                                006D   295
                                006D   296          .DSABL  LSB
```

```
                                006D   298          .SBTTL  EXECEPTION ENTRY POINT - PAGE TYPE DISPATCHER
                                006D   299
                                006D   300          .ALIGN  LONG
                                0070   301  MMG$PAGEFAULT::
                          55    DD    0070   302          PUSHL   R5                              ;SAVE R5
                          54    DD    0072   303          PUSHL   R4                              ;SAVE R4
        02   14 AE    05     10    ED    0074   304          CMPZV   #PSL$V_IPL,#PSL$S_IPL,<12+8>(SP),#IPL$_ASTDEL ;CHECK FAULT IPL
                               84    14    007A   305          BGTR    IPLHI                           ;BRANCH IF IPL IS TOO HIGH
                          53    DD    007C   306          PUSHL   R3                              ;SAVE R3
                                007E   307          SETIPL  #IPL$_SYNCH                     ;LOCK THE DATA BASE
                    12    08    DA    007E                MTPR    #IPL$_SYNCH,S^#PR$_IPL
                          52    DD    0081   308          PUSHL   R2                              ;SAVE R2
                54    0000'CF     D0    0083   309          MOVL    W^SCH$GL_CURPCB,R4              ;R4 = ADDRESS OF PCB
                          51    DD    0088   310          PUSHL   R1                              ;SAVE R1
                          50    DD    008A   311          PUSHL   R0                              ;SAVE R0
52   1C AE    000001FF  8F    CB    008C   312          BICL3   #^X1FF,FLTVA(SP),R2              ;R2 = VA OF FAULT (LOW BITS CLEAR)
                               AB    19    0095   313          BLSS    SYSTEMSPACE                     ;BRANCH IF SYSTEM SPACE ADDRESS
             55    6C    A4    D0    0097   314          MOVL    PCB$L_PHD(R4),R5                ;R5 = ADDRESS OF HEADER
          9D 36 A5     00    E0    009B   315          BBS     #PHD$V_PFMFLG,PHD$W_FLAGS(R5),PGFMONITOR ;BRANCH IF PAGE FAULT MONIT
                                00A0   316  PGFMONITOR1:
       53   52   15     09    EF    00A0   317          EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R3 ;VIRTUAL PAGE IN P0 OR P1 SPACE
            0B 52     1E    E1    00A5   318          BBC     #VA$V_P1,R2,P0ADDR              ;BRANCH IF P0 SPACE
          53    00D0 D543    DE    00A9   319          MOVAL   @PHD$L_P1BR(R5)[R3],R3          ;GET SYS VIRT ADR OF PTE FOR P1 SPACE
                          09    11    00AF   320          BRB     GETPAGELOC
                    0552    31    00B1   321  VALID:   BRW     PGFCOMPLETE                     ;IF VALID, JUST EXIT
          53    00C8 D543    DE    00B4   322  P0ADDR:  MOVAL   @PHD$L_P0BR(R5)[R3],R3          ;GET SYS VIRT ADR OF PTE FOR P0 SPACE
                                00BA   323  ;
                                00BA   324  ; R2 = VA (LOW BITS = PAGTYP), R3 = SVAPTE
                                00BA   325  ;
                                00BA   326  GETPAGELOC:
    50   53   15     09    EF    00BA   327          EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R0 ;INDEX TO SPT ENTRY
       51    0000'DF40    DE    00BF   328          MOVAL   @W^MMG$GL_SPTBASE[R0],R1 ;ADDRESS OF SPTE FOR PAGE TABLE
                          61    D5    00C5   329          TSTL    (R1)                            ;IS SPTE VALID?
                          9A    18    00C7   330          BGEQ    PPGTBL                          ;BRANCH IF NOT, FAULT IT
50   63    7B800000  8F    CB    00C9   331          BICL3   #^C<PTE$M_VALID                  -;CHECK VALID BIT
                                00D1   332                  ! PTE$M_TYP1 ! PTE$M_TYP0 -;GET PTE TYPE BITS
                                00D1   333                  ! PTE$M_PGFLVB>,(R3),R0 ;AND PFN/PAGE FILE VBN BITS TO R0
                    DE    19    00D1   334          BLSS    VALID                           ;BRANCH IF VALID
                                00D3   335  ;
                                00D3   336  ; R0 = TYP1 ! TYP0 ! PGFLVB,  VALID IS KNOWN TO BE OFF AT THIS POINT
                                00D3   337  ; R1 = SPT ENTRY ADDRESS FOR PAGE TABLE PAGE
                                00D3   338  ; R2 = VA (LOW BITS = PAGTYP)
                                00D3   339  ; R3 = SVAPTE
                                00D3   340  ;
                0000'CF     D6    00D3   341          INCL    W^PMS$GL_FAULTS                 ;COUNT ALL THE PAGE FAULTS
                  4C A5     D6    00D7   342          INCL    PHD$L_PAGEFLTS(R5)              ;COUNT PROCESS' PAGE FAULTS
                     53     DD    00DA   343          PUSHL   R3                              ;SAVE NEEDED VOLATILE REGISTERS
                     52     DD    00DC   344          PUSHL   R2
                     51     DD    00DE   345          PUSHL   R1
                   079C     30    00E0   346          BSBW    MMG$FREWSLE                     ;FREE A WORKING SET LIST ENTRY
                3F 50       E9    00E3   347          BLBC    R0,RSRCWAIT_3                   ;BRANCH IF HAVE TO WAIT
                                00E6   348  ;
                                00E6   349  ; ***** ALL POINTS DISPATCHED TO FROM HERE MUST REMEMBER THAT
                                00E6   350  ; ***** 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE
                                00E6   351  ;
             53    08 AE    D0    00E6   352          MOVL    8(SP),R3                        ;SVAPTE
                                00EA   353  ;
```

```
                                00EA   354 ; MUST RECHECK VALIDITY OF PAGE TABLE PAGE, SINCE FREWSLE MIGHT HAVE DISCARDED IT
                                00EA   355 ;
                          9E    D5  00EA   356         TSTL    @(SP)+                  ;IS PAGE TABLE PAGE VALID?
                          3D    18  00EC   357         BGEQ    PPGTBL_2                ;NO, SO FAULT THE PAGE TABLE
                                00EE   358 ;
                                00EE   359 ; MUST FETCH PAGE TABLE ENTRY CONTENTS AGAIN SINCE DEAD PAGE TABLE SCAN
                                00EE   360 ; IN FREWSLE MIGHT HAVE DISCARDED THE PAGE.  NOTE THAT VALID IS KNOWN 0.
                                00EE   361 ;
          50    63    7B800000  8F  CB  00EE   362         BICL3   #^C<PTE$M_VALID ! -     ;FETCH VALID BIT
                                00F6   363                 PTE$M_TYPT ! PTE$M_TYPO !- ;PTE TYPE BITS
                                00F6   364                 PTE$M_PGFLVB>,(R3),R0   ;AND PGFLVB, GPTX, SECX, PFN
                          2A    13  00F6   365         BEQL    DZRO_PTE_0              ;BRANCH IF PAGE IS DEMAND ZERO
                                00F8   366 ;
                                00F8   367 ; FORM R1 = 4 BIT SIGN EXTENDED VIELD LOW BIT = TYPO, SIGN = TYP1
                                00F8   368 ;
          51    50    05    16  EE  00F8   369         EXTV    #PTE$V_TYPO,#<PTE$V_TYP1+1-PTE$V_TYPO>,R0,R1
                          55    12  00FD   370         BNEQ    NOTRANSITION            ;BRANCH IF NOT TRANSITION PAGE
                                00FF   371 ;
                                00FF   372 ; THIS IS A PAGE IN TRANSITION
                                00FF   373 ; R0 = PFN, R2 = VA (LOW BITS = PAGTYP), R3 = SVAPTE
                                00FF   374 ;
                                00FF   375 TRANSITION:
    52  0000'DF40    03    00  EF  00FF   376         EXTZV   #PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0],R2 ;PAGE LOCATION
                                0107   377
                          00000002  0107   378         .IF     GT,CA$_MEASURE
                  0018'CF42    D6  0107   379         INCL    W^PMS$AL_TRANSFLT[R2]   ;COUNT VARIOUS TRANSITION FAULTS
                                010C   380         .ENDC
                                010C   381
                                010C   382         CASE    R2,<-
                                010C   383                 PFNLIST,-               ;ON THE FREE PAGE LIST
                                010C   384                 PFNLIST,-               ;ON THE MODIFIED PAGE LIST
                                010C   385                 PFNLIST,-               ;ON THE BAD PAGE LIST
                                010C   386                 RELEASEPEND,-           ;RELEASE PENDING
                                010C   387                 READERR,-               ;PAGE READ ERROR
                                010C   388                 WRITEINPROG,-           ;PAGE WRITE IN PROGRESS
                                010C   389                 READINPROG-             ;PAGE READ IN PROGRESS
                                010C   390             >
  06'   00    52    AF  CF  010C           CASEW   R2,#0,S^#<<30001$-30000$>/2>-1
                                0110       30000$:
                    04FC'       0110               .SIGNED_WORD    PFNLIST-30000$
                    04FC'       0112               .SIGNED_WORD    PFNLIST-30000$
                    04FC'       0114               .SIGNED_WORD    PFNLIST-30000$
                    04FF'       0116               .SIGNED_WORD    RELEASEPEND-30000$
                    054B'       0118               .SIGNED_WORD    READERR-30000$
                    04FF'       011A               .SIGNED_WORD    WRITEINPROG-30000$
                    0272'       011C               .SIGNED_WORD    READINPROG-30000$
                                011E       30001$:
                                011E   391 LOCBAD: BUG_CHECK PGFLOCBAD,FATAL       ;BAD PAGE LOCATION FIELD
                          FEFF  011E                       .WORD   ^XFEFF
                          0004' 0120                       .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_PGFLOCBAD!4
                                0122   392
                                0122   393 DZRO_PTE_0:
                      0409    31  0122   394         BRW     DZRO_PTE
                                0125   395 RSRCWAIT_3:
      5E    0C    C0  0125   396         ADDL    #3*4,SP                 ;CLEAN OFF 3 LONG WORDS
                0253    31  0128   397         BRW     RESOURCEWAIT            ;AND GO WAIT FOR A RESOURCE
                                012B   398 PPGTBL_2:
```

```
                03   BA  012B   399          POPR    #^M<R0,R1>              ;CLEAN OFF 2 LONG WORDS, SCRATCH R0,R1
            FF33    31  012D   400          BRW     PPGTBL                  ;FAULT A PROCESS PAGE TABLE
```

```
                                0130   402          .SBTTL  PAGE FILE, SECTION TABLE INDEX, OR GLOBAL PAGE
                                0130   403 ;
                                0130   404 ; GLOBAL PAGE, MASTER PTE VALID.
                                0130   405 ; R0 = MASTER PTE CONTENTS (VALID, MODIFY AND PFN BITS)
                                0130   406 ; 0(SP) = PROCESS VA (LOW BITS = PAGTYP), 4(SP) = SLAVE PTE ADDRESS
                                0130   407 ;
                                0130   408 GBLVALID:
                003C'CF    D6   0130   409          INCL    W^PMS$GL_GVALID         ;UPDATE GLOBAL VALID COUNTER
                52    6E   7D   0134   410          MOVQ    (SP),R2                 ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE
                     50    DD   0137   411          PUSHL   R0                      ;SAVE MASTER PTE
          50    50   15    00   EF   0139   412          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R0,R0 ;PAGE FRAME NUMBER
                   0950    30   013E   413          BSBW    MMG$MAKEWSLE            ;MAKE A WORKING SET LIST ENTRY
                     0E    BA   0141   414          POPR    #^M<R1,R2,R3>           ;R1=MASTER PTE, R2=VA, R3=SVAPTE
                   04FC    31   0143   415          BRW     SETSLAVEPTE             ;SET PROCESS' PTE AND EXIT
                                0146   416 ;
                                0146   417 ; DEMAND ZERO GLOBAL SECTION WITH PAGE FILE BACKING STORE
                                0146   418 ;
                                0146   419 GBLDZRO_PGFL:
          50    00400000 8F    D0   0146   420          MOVL    #PTE$M_TYP0,R0          ;ADD A TYP0 BIT TO THE MASTER PTE (0)
                   007A    31   014D   421          BRW     GBLDZRO                 ;GO JOIN COMMON CODE
                                0150   422 ;
                                0150   423 ; BAD MASTER PAGE TABLE ENTRY FORMAT FOR A GLOBAL PAGE
                                0150   424 ;
                                0150   425 GBLBAD: BUG_CHECK PGFGBLBAD,FATAL       ;BAD MASTER PTE FORMAT FOR GLOBAL PAGE
                        FEFF    0150                .WORD   ^XFEFF
                        0004'   0152                .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_PGFGBLBAD!4
                                0154   426 ;
                                0154   427 ; PAGE IS NOT A TRANSITION OR DEMAND ZERO PAGE
                                0154   428 ; R0 = LOW 23 BITS OF PTE AND TYPE BITS (PAGE FILE VBN, GPTX OR STX)
                                0154   429 ; R1 = RESULT OF EXTV ABOVE, CONDITION CODES SET FROM EXTV
                                0154   430 ; 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE
                                0154   431 ;
                                0154   432 NOTRANSITION:
                      7A    19   0154   433          BLSS    NOTGLOBAL               ;BRANCH IF TYP1 SET, NOT A GLOBAL PAGE
                00 50    16   E5   0156   434          BBCC    #PTE$V_TYP0,R0,10$      ;LEAVE JUST GLOBAL PAGE TABLE INDEX
             53    0000'DF40   DE   015A   435 10$:     MOVAL   @W^MMG$GL_GPTBASE[R0],R3 ;ADDRESS OF MASTER PTE
               51    63    0A   9C   0160   436          ROTL    #<32-<PTE$V_OWN-WSL$V_PAGTYP>>,(R3),R1 ;OWNER FIELD TO LOW BITS
                                0164   437 ;
                                0164   438 ; MASTER PTE OWNER FIELD CONTAINS THE VALUE PFN$C_GLOBAL OR PFN$C_GBLWRT
                                0164   439 ;
             6E    51    F1 8F    8B   0164   440          BICB3   #^C<WSL$M_PAGTYP>,R1,(SP) ;SET PAGE TYPE FIELD
          50   63    78800000 8F    CB   0169   441          BICL3   #^C<PTE$M_VALID ! -     ;GET THE VALID BIT
                                0171   442                  PTE$M_TYPT ! PTE$M_TYP0 ! - ;THE PTE TYPE BITS
                                0171   443                  PTE$M_PGFLVB>,(R3),R0   ;AND PAGE FILE VBN, OR STX
                      BD    19   0171   444          BLSS    GBLVALID                ;BRANCH IF MASTER PTE IS VALID
                      D1    13   0173   445          BEQL    GBLDZRO_PGFL            ;MASTER PTE IS DEMAND ZERO
          51    50    05    16   EE   0175   446          EXTV    #PTE$V_TYP0,#<PTE$V_TYP1+1-PTE$V_TYP0>,R0,R1
                                017A   447 ;
                                017A   448 ; R1 = 0 IF TYP1 AND TYP0 ARE BOTH ZERO
                                017A   449 ; R1 = NEGATIVE IF TYP1 IS SET
                                017A   450 ; R1<0> = 1 IF TYP0 IS SET
                                017A   451 ;
                      83    13   017A   452          BEQL    TRANSITION              ;BRANCH IF GLOBAL TRANSITION
                      D2    14   017C   453          BGTR    GBLBAD                  ;BRANCH IF GLOBAL AGAIN, ERROR
                                017E   454 ;
                                017E   455 ; MASTER PAGE TABLE ENTRY IS A SECTION OR PAGE FILE ADDRESS
                                017E   456 ;
```

```
                08 51   E9  017E   457 20$:    BLBC    R1,30$                          ;BRANCH IF PAGE FILE
             45 50   11   E0  0181   458        BBS     #PTE$V_DZRO,R0,GBLDZRO          ;BRANCH IF DEMAND ZERO GLOBAL SECTION
             06 50   10   E0  0185   459        BBS     #PTE$V_CRF,R0,GBLCRF            ;BRANCH IF COPY ON REFERENCE
                         0189   460 ;
                         0189   461 ; GLOBAL SECTION (NOT CRF OR DZRO) OR PAGE FILE BACKING STORE ADR
                         0189   462 ;
             50   04 AE   D0  0189   463 30$:    MOVL    4(SP),R0                        ;SAVE SLAVE PTE ADR INDICATING GLOBAL
                   4A   11  018D   464        BRB     GBLNOTRESIDENT                  ;
                         018F   465 ;
                         018F   466 ; GLOBAL COPY ON REFERENCE PAGE
                         018F   467 ;
                         018F   468 GBLCRF:
                   6E   94  018F   469        CLRB    (SP)                            ;SAY PAGE IS PROCESS PAGE
                50   53   D0  0191   470        MOVL    R3,R0                           ;MASTER PTE ADDRESS
                   43   11  0194   471        BRB     GBLNOTRESIDENT                  ;
```

```
                     0196   473          .SBTTL  PAGE NOT RESIDENT, QUEUE A READ REQUEST
                     0196   474          .ENABL  LSB
                     0196   475
                     0196   476 ;
                     0196   477 ; MUST WAIT FOR AN I/O REQUEST PACKET
                     0196   478 ;
                     0196   479 IRPWAIT_3:
          02   BA    0196   480          POPR    #^M<R1>                 ;CLEAN OFF 1 LONG WORD
    51    03   9A    0198   481          MOVZBL  #RSN$_NPDYNMEM,R1       ;NON PAGED DYNAMIC MEMORY RESOURCE NUMBER
          0C   BA    019B   482 10$:     POPR    #^M<R2,R3>              ;CLEAN OFF 2 LONG WORDS
       01DE    31    019D   483          BRW     RESOURCEWAIT            ;WAIT FOR RESOURCE IN R1
                     01A0   484 ;
                     01A0   485 ; THIS PAGE READ WOULD EXCEED THIS PROCESS' DIRECT I/O QUOTA.
                     01A0   486 ; WAIT UNTIL SOME OF HIS OUTSTANDING I/O COMPLETES.
                     01A0   487 ;
                     01A0   488 DIOCNTWAIT_2:
    51    01   9A    01A0   489          MOVZBL  #RSN$_ASTWAIT,R1        ;AST WAIT RESOURCE NUMBER
          F6   11    01A3   490          BRB     10$
                     01A5   491
                     01A5   492          .DSABL  LSB
                     01A5   493
                     01A5   494 ;
                     01A5   495 ; NO I/O PACKETS ON THE SIDE LIST, MUST ALLOCATE ONE FROM NON-PAGED POOL
                     01A5   496 ; 0(SP) = PLACE TO STORE ADDRESS OF PACKET, TOTAL OF 3 LONG WORDS ON STACK
                     01A5   497 ;
                     01A5   498 GET_IRP:
    6E    50   D0    01A5   499          MOVL    R0,(SP)                 ;SAVE REGISTER
51    C4 8F   9A    01A8   500          MOVZBL  #IRP$C_LENGTH,R1        ;SIZE OF I/O PACKET
     FE51'    30    01AC   501          BSBW    EXE$ALONONPAGED         ;ALLOCATE THE PACKET
     E4 50    E9    01AF   502          BLBC    R0,IRPWAIT_3            ;BRANCH IF NONE AVAILABLE
        01    BA    01B2   503          POPR    #^M<R0>                 ;RESTORE SAVED REGISTER
        52    DD    01B4   504          PUSHL   R2                      ;I/O PACKET ADDRESS
        6A    11    01B6   505          BRB     GOT_IRP                 ;REJOIN THE MAIN FLOW
                     01B8   506 ;
                     01B8   507 ; MUST WAIT FOR A FREE PAGE, 5 LONG WORDS ON STACK, FIRST 2 ARE GARBAGE
                     01B8   508 ; 8(SP) = I/O REQUEST PACKET ADDRESS TO BE DEALLOCATED, LAST 2 ARE SCRATCH
                     01B8   509 ;
                     01B8   510 FREPAGWAIT_5:
        03    BA    01B8   511          POPR    #^M<R0,R1>              ;SCRATCH R0,R1
        07    BA    01BA   512          POPR    #^M<R0,R1,R2>           ;I/O PACKET ADDRESS TO R0
08 A0 C4 8F   9A    01BC   513          MOVZBL  #IRP$C_LENGTH,IRP$W_SIZE(R0) ;SET PACKET SIZE AND CLEAR TYPE
     FE3C'    30    01C1   514          BSBW    EXE$DEANONPAGED         ;AND DEALLOCATE IT
     01B0     31    01C4   515          BRW     FREEPAGEWAIT
                     01C7   516 ;
                     01C7   517 ; DEMAND ZERO PROCESS SECTION PAGE
                     01C7   518 ;
                     01C7   519 DZRO_PROC_SEC:
        0364   31    01C7   520          BRW     DZRO_PTE
                     01CA   521 ;
                     01CA   522 ; DEMAND ZERO GLOBAL SECTION PAGE
                     01CA   523 ;
                     01CA   524 GBLDZRO:
    51    63   D0    01CA   525          MOVL    (R3),R1                 ;MASTER PTE CONTENTS
        0362   31    01CD   526          BRW     DZRO_GBL_SEC
                     01D0   527 ;
                     01D0   528 ; PAGE IS NOT A GLOBAL PAGE
                     01D0   529 ; R1<31>=TYP1, R1<0>=TYP0, R0 = TYP1 ! TYP0 ! PGFLVB
```

```
                                01D0    530 ;
                                01D0    531 NOTGLOBAL:
                          04 51 E9      01D0    532         BLBC    R1,10$                  ;BRANCH IF NOT SECTION PAGE
                       F0 50    11      01D3    533         BBS     #PTE$V_DZRO,R0,DZRO_PROC_SEC ;BRANCH IF DEMAND ZERO PROCESS SECTION
                             50 D4      01D7    534 10$:    CLRL    R0                      ;INDICATE NO SLAVE PAGE TABLE ENTRY
                                01D9    535 ;
                                01D9    536 ; 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE (SLAVE IF GLOBAL)
                                01D9    537 ; R0    = MASTER PTE ADDRESS IF GLOBAL CRF
                                01D9    538 ;       = SLAVE PTE ADDRESS IF GLOBAL NOT CRF
                                01D9    539 ;       = 0 IF NOT GLOBAL
                                01D9    540 ;
                                01D9    541 GBLNOTRESIDENT:
                       01   35 A5 91    01D9    542         CMPB    PHD$B_PGTBPFC(R5),#1    ;IF CLUSTERING PAGE TABLE PAGES
                             32 15      01DD    543         BLEQ    40$
                52   04 AE   15 09     01DF    544         EXTZV   #VA$V_VPN,#VA$S_VPN,4(SP),R2 ;SEE IF ADJACENT PAGE TABLES
                     51 0000'DF42 DE    01E5    545         MOVAL   @W^MMG$GL_SPTBASE[R2],R1 ;NEED TO BE FAULTED, GET SPT ENTRY ADR
                   71   847FFFFF 8F D3   01EB    546         BITL    #<PTE$M_VALID ! -
                                01F2    547                 PTE$M_TYP1 ! PTE$M_TYP0 ! -
                                01F2    548                 PTE$M_PGFLVB>,-(R1)     ;CHECK PREVIOUS SPT ENTRY
                             0E 14      01F2    549         BGTR    10$                     ;BRANCH IF NOT VALID, NOT DZRO
                08 A1   847FFFFF 8F D3   01F4    550         BITL    #<PTE$M_VALID ! -
                                01FC    551                 PTE$M_TYP1 ! PTE$M_TYP0 ! -
                                01FC    552                 PTE$M_PGFLVB>,8(R1)     ;CHECK NEXT SPT ENTRY
                             13 15      01FC    553         BLEQ    40$                     ;BRANCH IF IT IS VALID
                                01FE    554 ;
                                01FE    555 ; NEXT PAGE TABLE NEEDS TO BE FAULTED
                                01FE    556 ;
                             52 D6      01FE    557         INCL    R2                      ;SET NEXT SPT INDEX
                             02 11      0200    558         BRB     20$
                                0202    559 ;
                                0202    560 ; PREVIOUS PAGE TABLE NEEDS TO BE FAULTED
                                0202    561 ;
                             52 D7      0202    562 10$:    DECL    R2                      ;SET PREVIOUS SPT INDEX
                52   52      09 9C      0204    563 20$:    ROTL    #9,R2,R2                ;TURN SPT INDEX BACK INTO
                   00 52      1F E2     0208    564         BBSS    #VA$V_SYSTEM,R2,30$     ;SYSTEM VIRTUAL ADDRESS
                             03 BA      020C    565 30$:    POPR    #^M<R0,R1>              ;CLEAN STACK SCRATCH R0,R1
                          FE31 31       020E    566         BRW     SYSTEMSPACE             ;GO FAULT THE PAGE TABLE
                51   0000'CF D0         0211    567 40$:    MOVL    W^SCH$GL_CURPCB,R1      ;COULD HAVE SYSTEM PCB IN R4
                        3E A1 B5        0216    568         TSTW    PCB$W_DIOCNT(R1)        ;ENOUGH DIRECT I/O QUOTA FOR THIS READ?
                                0219    569                                             ;NOTE THAT BUILDPKT WILL CHARGE THE READ
                             85 15      0219    570         BLEQ    DIOCNTWAIT_2            ;BRANCH IF NO, MUST WAIT.
                7E   0000'DF   0F       021B    571         REMQUE  @W^IOC$GL_IRPFL,-(SP)   ;GET AN I/O PACKET FROM THE SIDE LIST
                             83 1D      0220    572         BVS     GET_IRP                 ;BRANCH IF NEED TO GET ONE FROM THE POOL
                                0222    573 ;
                                0222    574 ; R0    = MASTER PTE ADDRESS IF GLOBAL CRF
                                0222    575 ;       = SLAVE PTE ADDRESS IF GLOBAL NOT CRF
                                0222    576 ;       = 0 IF NOT GLOBAL
                                0222    577 ; 0(SP) = I/O REQUEST PACKET ADDRESS
                                0222    578 ;
                                0222    579 GOT_IRP:
                   52   04 AE   7D      0222    580         MOVQ    4(SP),R2                ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE
                             7E D4      0226    581         CLRL    -(SP)                   ;INIT CRF INDICATOR TO 'NOT CRF PAGE'
                             50 DD      0228    582         PUSHL   R0                      ;SAVE GLOBAL, GBLCRF INDICATOR
                          0822 30       022A    583         BSBW    MMG$ININEWPFN           ;ALLOCATE AND INIT A NEW PFN
                             50 D5      022D    584         TSTL    R0                      ;PFN ALLOCATED SUCCESSFULLY?
                             87 19      022F    585         BLSS    FREPAGWAIT_5            ;BRANCH IF NOT, MUST WAIT
                53   0000'DF40 D0       0231    586         MOVL    @W^PFN$AL_PTE[R0],R3    ;GET MASTER PTE ADDRESS
```

```
                                0237                                            ;SAME AS SLAVE UNLESS GLOBAL
            10 AE    53   D0    0237            MOVL    R3,16(SP)               ;SAVE FOR LATER USE
             0000'DF40    B6    023B            INCW    @W^PFN$AW_REFCNT[R0]    ;2ND REFERENCE FOR PAGE I/O
                                0240    ;
                                0240    ; FORM R2 = BACKING STORE ADDRESS
                                0240    ; 0(SP)  = MASTER PAGE TABLE ENTRY ADDRESS IF GLOBAL CRF
                                0240    ;        = SLAVE  PAGE TABLE ENTRY ADDRESS IF GLOBAL NOT CRF
                                0240    ;        = 0 IF NOT GLOBAL PAGE
                                0240    ; 4(SP)  = 0 INITIALIZED TO "NOT COPY ON REFERENCE"
                                0240    ;          SET THIS TO CORRECT BACKING STORE ADDRESS IF CRF PAGE
                                0240    ; 8(SP)  = I/O REQUEST PACKET ADDRESS
                                0240    ; 12(SP) = VIRTUAL ADDRESS (LOW BITS = PAGTYP)
                                0240    ; 16(SP) = SVAPTE, GLOBAL IF NOT GBL CRF, PROCESS IF NOT GBL OR IF GBL CRF
                                0240    ;
  51    63   867FFFFF 8F  CB    0240            BICL3   #^C<PTE$M_PROT ! PTE$M_OWN>,(R3),R1 ;R1 = PROT AND OWN
        52    63   51     CB    0248            BICL3   R1,(R3),R2              ;R2 = TYP1 ! TYP0 ! PGFLVB
        63    51   50     C9    024C            BISL3   R0,R1,(R3)              ;PTE = PROT ! OWN ! PFN = TRANSITION PTE
           14 52   1A     E4    0250            BBSC    #PTE$V_TYP1,R2,20$      ;BRANCH IF PAGE FILE OR SECTION
                                0254    ;
                                0254    ; GLOBAL COPY ON REFERENCE PAGE
                                0254    ;
              53   8E     D0    0254            MOVL    (SP)+,R3                ;GET MASTER PAGE TABLE ENTRY ADDRESS
  6E    52   00800000 8F  C9    0257            BISL3   #PFN$M_GBLBAK,R2,(SP)   ;GBL BACKING STORE ADR IN CRF INDICATOR
                   63     DD    025F            PUSHL   (R3)                    ;SAVE MASTER PAGE TABLE ENTRY CONTENTS
     52   63    17   00   EF    0261            EXTZV   #PFN$V_BAK,#PFN$S_BAK,(R3),R2 ;GET ADR FROM MASTER PTE
                     0C   11    0266            BRB     25$                     ;TO COPY ON REFERENCE SECTION LOGIC
                                0268    ;
                                0268    ; PAGE FILE OR SECTION ADDRESS
                                0268    ;
           18 52   16     E1    0268    20$:    BBC     #PTE$V_TYP0,R2,40$      ;BRANCH IF PAGING FILE
           24 52   10     E1    026C            BBC     #PTE$V_CRF,R2,50$       ;BRANCH IF NOT CRF
                                0270    ;
                                0270    ; COPY ON REFERENCE SECTION TABLE ENTRY
                                0270    ;
           04 AE   52     D0    0270            MOVL    R2,4(SP)                ;SAVE BACKING ADDRESS IN CRF INDICATOR
              1C A5       D0    0274    25$:    MOVL    PHD$L_PAGFIL(R5),-      ;NULL PAGE FILE ADDRESS
             0000'DF40          0277                    @W^PFN$AL_BAK[R0]       ;"NOT YET ALLOCATED" PAGING FILE ADR
              86 8F       88    027B            BISB    #<PFN$M_MODIFY ! PFN$C_RDINPROG>,-
             0000'DF40          027E                    @W^PFN$AB_STATE[R0]     ;FORCE MODIFY BIT, READ IN PROGRESS
                   1C     11    0282            BRB     60$
                                0284    ;
                                0284    ; PAGING FILE BACKING STORE ADDRESS
                                0284    ;
              51   55     D0    0284    40$:    MOVL    R5,R1
                   6E     D5    0287            TSTL    (SP)                    ;IS IT GLOBAL
                   05     13    0289            BEQL    45$                     ;NO
        51    0000'DF     DE    028B            MOVAL   @W^MMG$GL_SYSPHD,R1     ;GET SYSTEM HEADER
         52    1C A1      C8    0290    45$:    BISL    PHD$L_PAGFIL(R1),R2     ;SET PAGE FILE INDEX
                                0294    ;
                                0294    ; SECTION TABLE BACKING STORE ADDRESS
                                0294    ;
     0000'DF40     06     88    0294    50$:    BISB    #PFN$C_RDINPROG,@W^PFN$AB_STATE[R0] ;READ IN PROGRESS
     0000'DF40     52     D0    029A            MOVL    R2,@W^PFN$AL_BAK[R0]    ;STORE BACKING STORE ADDRESS
                                02A0    ;
                                02A0    ; R0 = PFN
                                02A0    ; R2 = BACKING STORE ADDRESS
                                02A0    ; R3 = PAGE TABLE ENTRY ADDRESS, PROCESS ADR IF NOT GLOBAL.
```

```
                                02A0    644 ;           GLOBAL ADDRESS IF GLOBAL OR GLOBAL CRF
                                02A0    645 ; R4 = PROCESS PCB IF PROCESS PAGE, PROCESS PAGE TABLE, OR GLOBAL PAGE,
                                02A0    646 ;    = SYSTEM PCB IF SYSTEM PAGE OR GLOBAL PAGE TABLE
                                02A0    647 ; R5 = PROCESS HEADER ADDRESS CORRESPONDING TO THE ABOVE PCB ADDRESS
                                02A0    648 ; 0(SP)   = MASTER PTE CONTENTS IF GLOBAL CRF (>0)
                                02A0    649 ;         = SLAVE PTE ADDRESS IF GLOBAL NOT CRF (<0)
                                02A0    650 ;         = 0 IF NOT GLOBAL
                                02A0    651 ; 4(SP)   = 0 IF PAGE IS NOT COPY ON REFERENCE
                                02A0    652 ;         = BACKING STORE ADDRESS (FOR GBL CRF TOO) IF CRF PAGE
                                02A0    653 ; 8(SP)   = I/O REQUEST PACKET ADDRESS
                                02A0    654 ; 12(SP)  = VIRTUAL ADDRESS (LOW BITS = PAGTYP)
                                02A0    655 ; 16(SP)  = SVAPTE, GLOBAL IF NOT GBL CRF, PROCESS IF NOT GBL OR IF GBL CRF
                                02A0    656 ;
          51    08 AE    D0    02A0    657 60$:    MOVL    8(SP),R1                ;ADDRESS OF I/O REQUEST PACKET
          3C A1    5D    D0    02A4    658         MOVL    FP,FP_SAV(R1)           ;SAVE A REGISTER
             5D    51    D0    02A8    659         MOVL    R1,FP                   ;USE THIS FOR CLUSTER CONTEXT
          40 AD    54    7D    02AB    660         MOVQ    R4,PCB_SAV(FP)          ;SAVE PCB, PHD ADDRESSES
                   6E    D5    02AF    661         TSTL    (SP)                    ;LESS THAN 0 IF GLOBAL PAGE
                   0E    13    02B1    662         BEQL    70$                     ;BRANCH IF NOT GLOBAL PAGE
                   07    14    02B3    663         BGTR    65$                     ;BRANCH IF GLOBAL CRF
    54    00000000'EF    DE    02B5    664         MOVAL   L^MMG$AL_SYSPCB,R4      ;USE SYSTEM PCB FOR GLOBAL PAGES
                                02BC    665                                         ;WANT THE PRIORITY FROM IT
    55    0000'DF    DE    02BC    666 65$:    MOVAL   @W^MMG$GL_SYSPHD,R5     ;SYSTEM PROCESS HEADER ADDRESS
                                02C1    667 ;
                                02C1    668 ; IF THE BACKING STORE ADDRESS IN R2 IS A GLOBAL ADDRESS, THEN R3 IS THE MASTER PTE
                                02C1    669 ; AND R5 IS THE SYSTEM PROCESS HEADER, OTHERWISE R3 IS THE PROCESS PTE
                                02C1    670 ; ADDRESS AND R5 IS THE PROCESS HEADER ADDRESS.
                                02C1    671 ;
          6D    52    7D    02C1    672 70$:    MOVQ    R2,PTEDAT(FP)           ;SAVE PTE DATA, AND ADDRESS
                                02C4    673
                                02C4    674         ASSUME  ASTPRM EQ AST+4
          10 AD    8E    7D    02C4    675         MOVQ    (SP)+,AST(FP)           ;STORE PARAMETERS TO IOPOST IN IRP
                                02C8    676 ;
                                02C8    677 ; FETCH THE TRANSFER PRIORITY FROM PROCESS PCB IF PROCESS PAGE,
                                02C8    678 ; PROCESS PAGE TABLE, OR GLOBAL CRF PAGE.  USE SYSTEM PRIORITY
                                02C8    679 ; IF SYSTEM PAGE, GLOBAL PAGE, OR GLOBAL PAGE TABLE.
                                02C8    680 ;
       23 AD    2F A4    90    02C8    681         MOVB    PCB$B_PRIB(R4),PRI(FP)  ;STORE PRIORITY OF TRANSFER IN IRP
    24 AD    0000'DF40    D0    02CD    682         MOVL    @W^PFN$AL_BAK[R0],BAK(FP) ;SAVE BACKING STORE ADDRESS
             78 8F    8B    02D4    683         BICB3   #^C<PFN$M_MODIFY ! PFN$M_LOC>,- ;AND STATE BYTE INFORMATION
    22 AD    0000'DF40             02D7    684                 @W^PFN$AB_STATE[R0],STATE(FP) ;FROM PFN DATA OF FIRST PAGE
                   6E    D4    02DD    685         CLRL    (SP)                    ;THROUGH WITH IRP ADDRESS, USE FOR SCRATCH
                                02DF    686                                         ;WILL BE PAGE TABLE FAULT CLUSTER IF PPGTBL
             48 AD    D4    02DF    687         CLRL    PHVREFCADR(FP)          ;ADDRESS OF PROCESS HEADER REF CNT IF PPGTBL
                                02E2    688
                                02E2    689         ASSUME  PFN$C_PROCESS EQ 0
                                02E2    690         ASSUME  PFN$C_SYSTEM  EQ 1
                                02E2    691         ASSUME  PFN$C_GLOBAL  EQ 2
                                02E2    692         ASSUME  PFN$C_GBLWRT  EQ 3
                                02E2    693         ASSUME  PFN$C_PPGTBL  EQ 4
                                02E2    694         ASSUME  PFN$C_GPGTBL  EQ 5
          03    01    ED    02E2    695         CMPZV   #WSL$V_PAGTYP,#WSL$S_PAGTYP,- ;IF PAGE TABLE PAGE
       04    04 AE             02E5    696                 4(SP),#PFN$C_PPGTBL     ;THEN SEPARATE CLUSTER FACTOR
                   1B    19    02E8    697         BLSS    90$                     ;BRANCH IF NOT PAGE TABLE
                   16    14    02EA    698         BGTR    80$                     ;BRANCH IF GLOBAL PAGE TABLE
                                02EC    699 ;
                                02EC    700 ; MUST RECORD A PROCESS HEADER REFERENCE FOR PAGE READ OF PROCESS HEADER PAGE
```

```
                                 02EC        ;
            51    42 A5    3C    02EC                MOVZWL  PHD$W_PHVINDEX(R5),R1        ;PROCESS HEADER VECTOR INDEX
       51    0000'DF41     3E    02F0                MOVAW   @W^PHV$GL_REFCBAS[R1],R1    ;ADDRESS OF PROCESS HEADER REF CNT
                     61    B6    02F6                INCW    (R1)                         ;COUNT ANOTHER REFERENCE
            48 AD    51    D0    02F8                MOVL    R1,PHVREFCADR(FP)            ;SAVE ADDRESS FOR CLUSTERING CODE
            6E    35 A5    90    02FC                MOVB    PHD$B_PGTBPFC(R5),(SP)       ;GET PAGE TABLE CLUSTER FACTOR
                     03    12    0300                BNEQ    90$                          ;BRANCH IF SPECIFIED
               6E    01    D0    0302        80$:    MOVL    #1,(SP)                      ;INDICATE NO CLUSTERING
                   08A7    30    0305        90$:    BSBW    MMG$INIBLDPKT                ;SET UP REGISTERS TO CALL BUILDPKT
  23 0000'CF  00000000'8F  E0    0308                BBS     #EXE$V_NOCLUSTER,W^EXE$GL_FLAGS,110$ ;BRANCH IF CLUSTERING DISABLED
            54    40 AD    7D    0312                MOVQ    PCB_SAV(FP),R4               ;RECOVER PCB, PHD ADDRESSES
                                 0316
                                 0316                ASSUME  SEC$B_PFC EQ PFL$B_PFC
       01 AE    0B A1      90    0316                MOVB    SEC$B_PFC(R1),1(SP)          ;PAGE FAULT CLUSTER FROM
                                 031B                                                     ;SECTION OR PAGE FILE CONTROL BLOCK
               51    6E    9A    031B                MOVZBL  (SP),R1                      ;SEE IF PAGE TABLE CLUSTER SPECIFIED
                     0A    14    031E                BGTR    100$                         ;BRANCH IF SPECIFIED
            51    01 AE    9A    0320                MOVZBL  1(SP),R1                     ;SEE IF PAGE FILE OR SECTION TABLE
                                 0324                                                     ;CLUSTER WAS SPECIFIED
                     04    14    0324                BGTR    100$                         ;BRANCH IF IT WAS
            51    34 A5    9A    0326                MOVZBL  PHD$B_DFPFC(R5),R1           ;DEFAULT FROM PROCESS HEADER
                                 032A                                                     ;PROCESS IF PROCESS OR GLOBAL PAGE
                                 032A                                                     ;SYSTEM IF SYSTEM PAGE
               01    51    D1    032A        100$:   CMPL    R1,#1                        ;CLUSTER OF 1?
                     06    15    032D                BLEQ    110$                         ;BRANCH IF YES
               5E    04    C0    032F                ADDL    #4,SP                        ;CLEAN OFF CLUSTER FACTOR SCRATCH
                   008E    31    0332                BRW     TRY_TO_CLUSTER               ;GO TRY TO CLUSTER
               51    01    D0    0335        110$:   MOVL    #1,R1                        ;ONE PAGE READ
               5E    08    C0    0338                ADDL    #8,SP                        ;CLEAN OFF VIRTUAL ADDRESS
               53    8E    D0    033B                MOVL    (SP)+,R3                     ;SVAPTE FROM PFN$AL_PTE
                                 033E        ;
                                 033E        ; R0 = VBN, R1 = PAGE COUNT, R2 = WINDOW, R3 = SVAPTE, FP = IRP
                                 033E        ;
                                 033E        QUEUE_PAGE_READ:
                                 033E
                  0008'CF  D6    033E                INCL    W^PMS$GL_PREADIO             ;COUNT PAGE READ I/O REQUESTS (SYSTEM)
          0004'CF    51    C0    0342                ADDL    R1,W^PMS$GL_PREADS           ;AND THE NUMBER OF PAGES READ
          54    0000'CF    D0    0347                MOVL    W^SCH$GL_CURPCB,R4           ;PCB ADDRESS
            55    6C A4    D0    034C                MOVL    PCB$L_PHD(R4),R5             ;PHD ADDRESS
                0108 C5    D6    0350                INCL    PHD$L_PGFLTIO(R5)            ;COUNT PAGE READ I/O REQUESTS (PROCESS)
         51    51    09    78    0354                ASHL    #9,R1,R1                     ;FORM BYTE COUNT TO TRANSFER
               55    5D    D0    0358                MOVL    FP,R5                        ;I/O PACKET ADDRESS
            5D    3C A5    D0    035B                MOVL    FP_SAV(R5),FP                ;RESTORE SAVED REGISTER
                                 035F
                                 035F        ;
                                 035F        ; R0 = VBN
                                 035F        ; R1 = NUMBER OF BYTES TO READ
                                 035F        ; R2 = WINDOW ADDRESS
                                 035F        ; R3 = SVAPTE (MASTER IF GLOBAL, SLAVE IF GLOBAL CRF)
                                 035F        ; R4 = PROCESS PCB ADDRESS
                                 035F        ; R5 = I/O REQUEST PACKET ADDRESS
                                 035F        ;
                                 035F        ; IRP$L_AST(R5)
                                 035F        ;     = MASTER PTE CONTENTS IF GLOBAL CRF (>0)
                                 035F        ;     = SLAVE PTE ADDRESS IF GLOBAL NOT CRF (<0)
                                 035F        ;     = 0 IF NOT GLOBAL
                                 035F        ; IRP$L_ASTPRM(R5)
```

```
                        035F   758 ;       = BACKING STORE ADDRESS IF CRF PAGE (GBLBAK SET IF GBL CRF)
                        035F   759 ;       = 0 IF NOT CRF PAGE
                        035F   760 ; IRP$B_PRI(R5) = DESIRED TRANSFER PRIORITY
                        035F   761 ;
                FC9E'   30      035F   762         BSBW    EXE$BUILDPKTR           ;BUILD AND QUEUE THE I/O PACKET
                        0362   763 ;
                        0362   764 ; THE FOLLOWING WAITS THE PROCESS AT THE FAULTING MODE
                        0362   765 ;
                        0362   766 PROCPAG:
    50  0000'CF    7E   0362   767         MOVAQ   W^SCH$GQ_PFWQ,R0        ;PAGE FAULT WAIT QUEUE ADDRESS
                49  10   0367   768         BSBB    MMG$PGFLTWAIT_1         ;PUT PROCESS ON PAGE FAULT WAIT QUEUE
                        0369   769 PGFEXIT:
                3F  BA   0369   770         POPR    #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE REGISTERS SAVED BY PAGE FAULT
                5E  08   C0    036B   771         ADDL    #8,SP                   ;CLEAN OFF THE EXCEPTION PARAMETERS
                        036E   772 ;
                        036E   773 ; STACK NOW CONTAINS JUST THE FAULT PC, PSL PAIR
                        036E   774 ;
                        036E   775 MMG$SVPCTX::
                07      036E   776         SVPCTX                          ;SAVE PROCESS CONTEXT
    54  0000'CF    D0   036F   777         MOVL    W^SCH$GL_CURPCB,R4      ;GET PCB ADDRESS
                FC89'   31      0374   778         BRW     SCH$WAITR               ;JOIN COMMON WAIT CODE FOLLOWING SVPCTX
                        0377   779 ;
                        0377   780 ; NO FREE PAGES AVAILABLE ON THE FREE PAGE LIST, MUST WAIT
                        0377   781 ;
                        0377   782 FREEPAGEWAIT:
    50  0000'CF    7E   0377   783         MOVAQ   W^SCH$GQ_FPGWQ,R0       ;WAIT ON FREE PAGE WAIT QUEUE
                1A  11   037C   784         BRB     FREEPAGEWAIT1
                        037E   785 ;
                        037E   786 ; WAIT FOR RESOURCE IN R1 TO BECOME AVAILABLE
                        037E   787 ;
                        037E   788 RESOURCEWAIT:
                1C  10   037E   789         BSBB    MMG$RESRCWAIT           ;SET UP TO WAIT FOR THE RESOURCES
                E7  11   0380   790         BRB     PGFEXIT                 ;AND EXIT TO THE SCHEDULER
                        0382   791 ;
                        0382   792 ; FAULT FOR PAGE WHICH IS ALREADY ON THE WAY INTO MEMORY
                        0382   793 ;
                        0382   794 READINPROG:
                0C  BA   0382   795         POPR    #^M<R2,R3>              ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE
                        0384   796         ASSUME  PFN$C_PROCESS EQ 0
    0000'DF40   07  93   0384   797         BITB    #PFN$M_PAGTYP,@W^PFN$AB_TYPE[R0] ;PROCESS PAGE?
                D6  13   038A   798         BEQL    PROCPAG                 ;BRANCH IF YES
00  0000'DF40   04  E2   038C   799         BBSS    #PFN$V_COLLISION,@W^PFN$AB_TYPE[R0],10$ ;COLLISION OCCURRED
                        0393   800 10$:
    50  0000'CF    7E   0393   801         MOVAQ   W^SCH$GQ_COLPGWQ,R0     ;COLLISION PAGE WAIT QUEUE
                        0398   802 FREEPAGEWAIT1:
                18  10   0398   803         BSBB    MMG$PGFLTWAIT_1         ;PLACE PROCESS ON THE COLLISION QUEUE
                CD  11   039A   804         BRB     PGFEXIT                 ;EXIT TO THE SCEDULER
                        039C   805 ;
                        039C   806 ; WAIT FOR RESOURCE IN R1 TO BECOME AVAILABLE
                        039C   807 ; R0,R1,R2,R3 ALTERED, R4 RETURNED WITH CURRENT PCB ADDRESS
                        039C   808 ;
                        039C   809         .ENABL  LSB
                        039C   810 MMG$RESRCWAIT::
    54  0000'CF    D0   039C   811         MOVL    W^SCH$GL_CURPCB,R4      ;R4 = CURRENT PCB ADDRESS
      4C A4     51  D0   03A1   812         MOVL    R1,PCB$L_EFWM(R4)       ;SET RESOURCE NEEDED
00  0000'CF    51  E6   03A5   813         BBSSI   R1,W^SCH$GL_RESMASK,10$ ;NOTE SOMEONE WAITING
                        03AB   814 10$:
```

```
                50   0000'CF   7E   03AB   815          MOVAQ   W^SCH$GQ_MWAIT,R0       ;WAIT ON MISCELLANEOUS QUEUE
                      05   11   03B0   816          BRB     20$                      ;GO WAIT THIS PROCESS
                                03B2   817 MMG$PGFLTWAIT_1:
                54   0000'CF   D0   03B2   818          MOVL    W^SCH$GL_CURPCB,R4       ;MUST WAIT THE PROCESS PCB
                                03B7   819 MMG$PGFLTWAIT::
                                03B7   820 20$:
                     08 A0   B6   03B7   821          INCW    WQH$W_WQCNT(R0)          ;COUNT THIS PROCESS WAITING
                60      64   0E   03BA   822          INSQUE  (R4),(R0)                ;QUEUE THIS PCB
              2C A4    0A A0   B0   03BD   823          MOVW    WQH$W_WQSTATE(R0),PCB$W_STATE(R4) ;SET WAIT STATE IN PCB
                                05   03C2   824          RSB
                                     03C3   825
                                     03C3   826          .DSABL  LSB
```

```
                                03C3    828          .SBTTL  FORM A CLUSTER OF PAGES TO READ
                                03C3    829 ;
                                03C3    830 ; R0 = VBN IN FILE OF FIRST PAGE TO READ
                                03C3    831 ; R1 = DESIRED CLUSTER SIZE
                                03C3    832 ; R2 = WINDOW CONTROL BLOCK ADDRESS
                                03C3    833 ; R4 = PCB ADDRESS, PROCESS IF PROCESS OR GLOBAL PAGE, SYSTEM IF SYSTEM PAGE
                                03C3    834 ; R5 = PHD ADDRESS, PROCESS IF PROCESS OR GLOBAL PAGE, SYSTEM IF SYSTEM PAGE
                                03C3    835 ; FP = I/O REQUEST PACKET ADDRESS
                                03C3    836 ; 0(SP) = VIRTUAL ADDRESS (LOW BITS = PAGTYP)
                                03C3    837 ; 4(SP) = SVAPTE FROM PFN$AL_PTE
                                03C3    838 ;
                                03C3    839
                                03C3    840          .ENABL  LSB
                                03C3    841
                                03C3    842 TRY_TO_CLUSTER:
                0C AD   8E   D0 03C3    843          MOVL    (SP)+,VA(FP)            ;SAVE VIRTUAL ADDRESS
                34 AD   50   D0 03C7    844          MOVL    R0,VBN(FP)              ;SET VIRTUAL BLOCK NUMBER
                                03CB    845
                                03CB    846          ASSUME  COUNT EQ CLUSTER+1
            20 AD   51 0100 8F   A9 03CB    847          BISW3   #^X0100,R1,CLUSTER(FP)  ;SET COUNT AND CLUSTER
                38 AD   52   D0 03D2    848          MOVL    R2,WINDOW(FP)           ;WINDOW ADDRESS
                                03D6    849 ;
                                03D6    850 ; PUT PTEDAT INTO FORM OF TYP1 ! TYP0 ! PGFLVB
                                03D6    851 ;
                                03D6    852          ASSUME  PFN$V_PGFLX GE 24
                                03D6    853          ASSUME  PTE$V_TYP1 GE 24
                03 AD   04   90 03D6    854          MOVB    #PTE$M_TYP1@-24,PTEDAT+3(FP) ;TURN TYP1 BACK ON, CLEAR PAGE FILE IND
                53   10 AD   D0 03DA    855          MOVL    AST(FP),R3              ;PROCESS PTE ADR IF GBL NOT CRF
                        18   13 03DE    856          BEQL    30$                     ;BRANCH IF NOT GLOBAL PAGE
                        0C   19 03E0    857          BLSS    10$                     ;BRANCH IF GBL NOT CRF
                                03E2    858 ;
                                03E2    859 ; GLOBAL COPY ON REFERENCE PAGE
                                03E2    860 ;
                    53   6E   D0 03E2    861          MOVL    (SP),R3                 ;PROCESS PTE ADR WHEN GBL CRF
                52   14 AD   D0 03E5    862          MOVL    ASTPRM(FP),R2           ;GPTX PTE CONTENTS FOR THIS CASE
                        06   11 03E9    863          BRB     20$
                                03EB    864
                                03EB    865 CLU_END1:
                      00EB   31 03EB    866          BRW     CLU_END
                                03EE    867
                                03EE    868 ;
                                03EE    869 ; GLOBAL PAGE NOT COPY ON REFERENCE
                                03EE    870 ;
                    52   63   D0 03EE    871 10$:     MOVL    (R3),R2                 ;GPTX FROM PROCESS PTE
            52  FB800000 8F   CA 03F1    872 20$:     BICL    #^C<PTE$M_TYP1 ! PTE$M_TYP0 !- ;ISOLATE PAGE TYPE
                                03F8    873                  PTE$M_GPTX>,R2          ;AND GPTX BITS
                                03F8    874
                                03F8    875          ASSUME  GPTX_PTE EQ GPTX+4
                18 AD   52   7D 03F8    876 30$:     MOVQ    R2,GPTX(FP)             ;SET GPTX AND GPTX_PTE
                28 AD   01   D0 03FC    877          MOVL    #1,INC1(FP)             ;INIT TO SCAN FORWARDS
                                0400    878
                                0400    879          .DSABL  LSB
                                0400    880
                                0400    881 CLU_INI_INC:
        2C AD   28 AD   02   78 0400    882          ASHL    #2,INC1(FP),INC4(FP)    ;* OR - 4
        30 AD   28 AD   09   78 0406    883          ASHL    #9,INC1(FP),INC512(FP)  ;* OR - 512
                                040C    884 CLU_NXT:
```

```
   53   04 AD   2C AD   C1  040C           ADDL3   INC4(FP),SVAPTE(FP),R3  ;NEXT PTE TO CHECK
          04 AD   53    D0  0412           MOVL    R3,SVAPTE(FP)            UPDATE CONTEXT
          04 6D   16    E0  0416           BBS     #PTE$V_TYPO,PTEDAT(FP),20$ ;BRANCH IF SECTION ADDRESS
          6D    28 AD   C0  041A           ADDL    INC1(FP),PTEDAT(FP)     ;INCREMENT PAGE FILE ADDRESS
   51    53   15   09   EF  041E    20$:   EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;CHECK THAT PTE IS RESIDENT
            0000'DF41   D5  0423           TSTL    @W^MMG$GL_SPTBASE[R1]   ;BY MAKING SURE ITS SPTE IS VALID
                   C1   18  0428           BGEQ    CLU_END1                ;BRANCH IF IT ISN'T
50   63    7B800000 8F  CB  042A           BICL3   #^C<PTE$M_VALID !-      ;GET VALID BIT
                            0432                   PTE$M_TYPT ! PTE$M_TYPO ! -;PAGE TYPE BITS
                            0432                   PTE$M_PGFLVB>,(R3),R0   ;AND PAGE FILE/GPTX BITS FROM PTE
             6D    50   D1  0432           CMPL    R0,PTEDAT(FP)           ;MUST AGREE IF THIS PAGE IS IN THE CLUSTER
                   B4   12  0435           BNEQ    CLU_END1                ;BRANCH IF AT END OF CLUSTER
                1C AD   D5  0437           TSTL    GPTX_PTE(FP)            ;WAS THAT THE MASTER PTE FOR A GLOBAL?
                   29   13  043A           BEQL    60$                     ;BRANCH IF NO, IT WAS PROCESS PTE
                            043C   ;
                            043C   ; MUST TEST THAT PROCESS PTE POINTS AT THE GPTX
                            043C   ;
   53   1C AD   2C AD   C1  043C           ADDL3   INC4(FP),GPTX_PTE(FP),R3 ;NEXT PROCESS PTE ADR
   51    53   15   09   EF  0442           EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;CHECK THAT THIS PTE IS ACCESSIBLE
            0000'DF41   D5  0447           TSTL    @W^MMG$GL_SPTBASE[R1]   ;BY MAKING SURE ITS SPTE IS VALID
                   9D   18  044C           BGEQ    CLU_END1                ;BRANCH IF IT ISN'T
50   63    7B800000 8F  CB  044E           BICL3   #^C<PTE$M_VALID ! -     ;GET VALID BIT
                            0456                   PTE$M_TYPT ! PTE$M_TYPO ! -;PAGE TYPE BITS
                            0456                   PTE$M_PGFLVB>,(R3),R0   ;AND PGFLVB/GPTX FROM PTE
   52   18 AD   28 AD   C1  0456           ADDL3   INC1(FP),GPTX(FP),R2    ;NEXT GLOBAL PAGE TABLE INDEX
             52    50   D1  045C           CMPL    R0,R2                   ;IN THE CLUSTER?
                   8A   12  045F           BNEQ    CLU_END1                ;BRANCH IF NOT
                            0461
                            0461           ASSUME  GPTX_PTE EQ GPTX+4
          18 AD    52   7D  0461           MOVQ    R2,GPTX(FP)             ;UPDATE GPTX
                            0465   ;
                            0465   ; R1 = SPT INDEX FOR PAGE TABLE PAGE
                            0465   ; R3 = PROCESS PTE ADDRESS
                            0465   ;
                   0A   BB  0465    60$:   PUSHR   #^M<R1,R3>              ;SAVE PROCESS PTE ADR, SPT INDEX
                 0415   30  0467           BSBW    MMG$FREWSLE             ;GET A FREE WORKING SET LIST ENTRY
             52    8E   7D  046A           MOVQ    (SP)+,R2                ;RESTORE PROCESS PTE ADR, SPT INDEX
                67 50   E9  046D           BLBC    R0,CLU_END_RESRC1       ;IF CANNOT GET ONE, END THE CLUSTER
                            0470   ;
                            0470   ; MUST CHECK THE SPT ENTRY FOR PROCESS PAGE TABLE IS STILL VALID
                            0470   ; FREWSLE MIGHT HAVE DISCARDED IT FROM THE WORKING SET.
                            0470   ;
            0000'DF42   D5  0470           TSTL    @W^MMG$GL_SPTBASE[R2]   ;IS SPT ENTRY FOR PT STILL VALID
                   7C   18  0475           BGEQ    CLU_END_RESRC           ;BRANCH IF NOT
   52   0C AD   30 AD   C1  0477           ADDL3   INC512(FP),VA(FP),R2    ;NEXT VIRTUAL ADDRESS
          0C AD    52   D0  047D           MOVL    R2,VA(FP)               ;UPDATE THE CONTEXT
                 05CB   30  0481           BSBW    MMG$ININEWPFN           ;ALLOC AND INIT A PFN
                   50   D5  0484           TSTL    R0                      ;IF NO PFN'S AVAILABLE
                   6B   19  0486           BLSS    CLU_END_RESRC           ;THEN END THE CLUSTER
                21 AD   96  0488           INCB    COUNT(FP)               ;COUNT ANOTHER PAGE IN THE CLUSTER
         51     48 AD   D0  048B           MOVL    PHVREFCADR(FP),R1       ;PROCESS HEADER REF CNT ADR
                            048F                                           ;IF THIS IS A PROCESS PAGE TABLE PAGE
                   02   13  048F           BEQL    70$                     ;BRANCH IF NOT A PROCESS PAGE TABLE PAGE
                   61   B6  0491           INCW    (R1)                    ;COUNT ANOTHER PROCESS HEADER REFERENCE
   53       0000'DF40   D0  0493    70$:   MOVL    @W^PFN$AL_PTE[R0],R3    ;PROCESS PTE ADR IF NOT GLOBAL OR IF GBL CRF
                            0499                                           ;GLOBAL PTE ADR IF GLOBAL NOT CRF
            0000'DF40   B6  0499           INCW    @W^PFN$AW_REFCNT[R0]    ;SECOND REFERENCE FOR I/O IN PROGRESS
```

```
 51   63   867FFFFF 8F   CB  049E           BICL3   #^C<PTE$M_PROT ! PTE$M_OWN>,(R3),R1 ;PROTECTION AND OWNER FIELDS
      63   51       50   C9  04A6           BISL3   R0,R1,(R3)               ;FORM TRANSITION PTE FORMAT
           52    24 AD   D0  04AA           MOVL    BAK(FP),R2               ;BACKING STORE FROM PREV PFN
           0E 52    16   E0  04AE           BBS     #PTE$V_TYP0,R2,80$       ;BRANCH IF SECTION ADDRESS
      51   52       09   78  04B2           ASHL    #32-PFN$S_BAK,R2,R1      ;IF NOT A NULL PAGE FILE ADDRESS
                    08   13  04B6           BEQL    80$
           52    28 AD   C0  04B8           ADDL    INC1(FP),R2              ;THEN INCREMENT THE ADDRESS
           24 AD    52   D0  04BC           MOVL    R2,BAK(FP)               ;AND UPDATE THE CONTEXT
      0000'DF40     52   D0  04C0  80$:     MOVL    R2,@W^PFN$AL_BAK[R0]     ;SET BACKING STORE ADR FOR THIS PFN
   0000'DF40     22 AD   88  04C6           BISB    STATE(FP),@W^PFN$AB_STATE[R0] ;USE STATE FROM PREV PFN
      20 AD      21 AD   91  04CD           CMPB    COUNT(FP),CLUSTER(FP)    ;IS CLUSTER FULL?
                    1F   18  04D2           BGEQ    CLU_END_RESRC            ;BRANCH IF YES, QUEUE THE READ
                  FF35   31  04D4           BRW     CLU_NXT                  ;NO, TRY FOR ANOTHER PAGE
                             04D7  CLU_END_RESRC1:
                    1A   11  04D7           BRB     CLU_END_RESRC
                             04D9  ;
                             04D9  ; END OF CLUSTER
                             04D9  ;
                             04D9  CLU_END:
           01    21 AD   91  04D9           CMPB    COUNT(FP),#1             ;IF AT LEAST 2 PAGES IN CLUSTER
                    14   14  04DD           BGTR    CLU_END_RESRC            ;THEN READ THE CLUSTER
      28 AD      28 AD   CE  04DF           MNEGL   INC1(FP),INC1(FP)        ;OTHERWISE TRY TO SCAN BACKWARDS
                    0D   14  04E4           BGTR    CLU_END_RESRC            ;UNLESS ALREADY TRIED THAT
           04 AD    04   C2  04E6           SUBL    #4,SVAPTE(FP)            ;BACK TO STARTING SVAPTE
           02 6D    16   E0  04EA           BBS     #PTE$V_TYP0,PTEDAT(FP),20$ ;BRANCH IF SECTION PAGE
                    6D   D7  04EE           DECL    PTEDAT(FP)               ;BACK TO ORIG PAGE FILE VBN
                  FF0D   31  04F0  20$:     BRW     CLU_INI_INC
                             04F3  ;
                             04F3  ; SET UP TO DO THE PAGE READ
                             04F3  ;
                             04F3  CLU_END_RESRC:
           53       8E   D0  04F3           MOVL    (SP)+,R3                 ;GET PTE ADR OF FIRST PFN IN CLUSTER
      50         34 AD   D0  04F6           MOVL    VBN(FP),R0               ;AND ITS ASSOCIATED VBN IN THE FILE
      51         21 AD   9A  04FA           MOVZBL  COUNT(FP),R1             ;NUMBER OF PAGES IN THE CLUSTER
                 28 AD   D5  04FE           TSTL    INC1(FP)                 ;IF CLUSTER WENT BACKWARDS
                    1E   14  0501           BGTR    20$
      52   01       51   C3  0503           SUBL3   R1,#1,R2                 ;-(COUNT-1)
           50       52   C0  0507           ADDL    R2,R0                    ;ADJUST FILE VBN
           53     6342   DE  050A           MOVAL   (R3)[R2],R3              ;AND PTE ADR
      54         10 AD   D0  050E           MOVL    AST(FP),R4               ;PROCESS PTE ADDRESS FOR GLOBAL PAGE?
                    0D   13  0512           BEQL    20$                      ;BRANCH IF NOT GLOBAL
                    06   19  0514           BLSS    10$                      ;BRANCH IF GLOBAL BUT NOT CRF
           14 AD    52   C0  0516           ADDL    R2,ASTPRM(FP)            ;ADJUST GLOBAL BACKING STORE ADDRESS
                    05   11  051A           BRB     20$
      10 AD       6442   DE  051C  10$:     MOVAL   (R4)[R2],AST(FP)         ;ADJUST PROCESS PTE ADDRESS FOR GLOBAL PAGE
      52         38 AD   D0  0521  20$:     MOVL    WINDOW(FP),R2            ;GET WINDOW ADDRESS
                  FE16   31  0525           BRW     QUEUE_PAGE_READ          ;GO QUEUE THE PAGE READ
```

B 7

```
                          0528          .SBTTL  DEMAND ZERO PAGE
                          0528  ;
                          0528  ; MUST WAIT FOR FREE PAGES TO BECOME AVAILABLE
                          0528  ;
                          0528  DZROFPGWAIT_5:
        5E    14    C0    0528          ADDL    #<5*4>,SP              ;CLEAN OFF 5 LONG WORDS
            FE49    31    052B          BRW     FREEPAGEWAIT           ;AND GO WAIT FOR A FREE PAGE
                          052E  ;
                          052E  ; THIS IS A DEMAND ZERO FORMAT PAGE TABLE ENTRY, R0 = 0
                          052E  ;
                          052E  DZRO_PTE:
              53    D4    052E          CLRL    R3                     ;NO GLOBAL MASTER PTE ADDRESS
              51    D4    0530          CLRL    R1                     ;NO MASTER PTE CONTENTS
                          0532  ;
                          0532  ; R0 = BACKING STORE ADDRESS, TYP1 ! TYP0 ! PGFLVB
                          0532  ; R1 = MASTER PAGE TABLE ENTRY CONTENTS IF GLOBAL, 0 IF NOT
                          0532  ; R3 = GLOBAL PAGE TABLE ENTRY ADDRESS IF GLOBAL, 0 IF NOT
                          0532  ; 0(SP) = FAULT VA (LOW BITS = PAGTYP)
                          0532  ; 4(SP) = CORRESPONDING SVAPTE
                          0532  ;
                          0532  DZRO_GBL_SEC:
              0B    BB    0532          PUSHR   #^M<R0,R1,R3>          ;SAVE GBL PTE ADR, GBL PTE CONTENTS,
                          0534                                         ;AND BACKING STORE ADDRESS
        52 0C AE    7D    0534          MOVQ    12(SP),R2              ;R2=VA, R3=SVAPTE
            0514    30    0538          BSBW    MMG$ININEWPFN          ;ALLOCATE AND INIT A PFN
              50    D5    053B          TSTL    R0                     ;SEE IF A PFN WAS ALLOCATED
              E9    19    053D          BLSS    DZROFPGWAIT_5          ;BRANCH IF HAVE TO WAIT
  0000'DF40   07    88    053F          BISB    #PFN$C_ACTIVE,@W^PFN$AB_STATE[R0] ;MARK PAGE ACTIVE
        52 1C A5    D0    0545          MOVL    PHD$L_PAGFIL(R5),R2    ;ASSUME NULL PAGE FILE BACKING STORE
           51 8E    D0    0549          MOVL    (SP)+,R1               ;GET BACKING STORE ADDRESS
              1F    13    054C          BEQL    20$                    ;BRANCH IF DEMAND ZERO FORMAT
     1B 51    10    E0    054E          BBS     #PTE$V_CRF,R1,20$      ;BRANCH IF DZRO, CRF SECTION
     0F 51    1A    E0    0552          BBS     #PTE$V_TYP1,R1,10$     ;CHECK FOR GLOBAL WITH PAGE FILE BACING STOR
     0B 51    16    E5    0556          BBCC    #PTE$V_TYP0,R1,10$
                          055A  ; PAGE FILE BACKING STORE GLOBAL SECTION
     52   0000'DF   DE    055A          MOVAL   @W^MMG$GL_SYSPHD,R2    ;GET SYSTEM HEADER
        52 1C A2    D0    055F          MOVL    PHD$L_PAGFIL(R2),R2    ;BACKING STORE ADDRESS
              08    11    0563          BRB     20$
52  51  FF820000 8F CB    0565  10$:    BICL3   #<PTE$M_DZRO ! ^C<PFN$M_BAK>>,R1,R2 ;BACKING STORE ADR
                          056D                                         ;WITH DZRO SHUT OFF
  0000'DF40   52    D0    056D  20$:    MOVL    R2,@W^PFN$AL_BAK[R0]   ;STORE THE BACKING STORE ADDRESS
                          0573  ;
                          0573  ;  0(SP) = MASTER PTE CONTENTS IF GLOBAL, 0 IF NOT
                          0573  ;  4(SP) = MASTER PTE ADDRESS  IF GLOBAL, 0 IF NOT
                          0573  ;  8(SP) = VIRTUAL ADDRESS (LOW BITS = PAGE TYPE)
                          0573  ; 12(SP) = SYSTEM VIRTUAL ADDRESS OF PROCESS PAGE TABLE ENTRY
                          0573  ;
        52 08 AE    7D    0573          MOVQ    8(SP),R2               ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE
    63   867FFFFF 8F CB   0577          BICL3   #^C<PTE$M_PROT ! PTE$M_OWN>, -
              7E          057E                  (R3),-(SP)             ;PROTECTION AND OWNER FROM PTE
63  50  94000000 8F C9    057F          BISL3   #<PTE$M_VALID ! PTE$C_KW ! -
                          0587                  PTE$M_MODIFY>,R0,(R3)  ;MAKE PAGE KERNEL WRITE FOR ZEROING
     7E   53    55  C3    0587          SUBL3   R5,R3,-(SP)            ;SAVE BYTE INDEX TO PTE
     03   01        ED    058B          CMPZV   #WSL$V_PAGTYP,#WSL$S_PAGTYP,- ;IF THIS IS NOT A PROCESS PAGE
     00   52              058E                  R2,#PFN$C_PROCESS      ;THEN DON'T LOWER IPL TO ZERO PAGE
          03        12    0590          BNEQ    22$                    ;STAY AT HIGHER IPL IF NOT PROCESS PAGE
                          0592
```

```
                                0592  1047          SETIPL  #IPL$_ASTDEL                ;SWAPPABLE WHILE ZEROING THE PAGE
                     12  02  DA  0592                MTPR    #IPL$_ASTDEL,S^#PR$_IPL
                                0595  1048
                                0595  1049 22$:
                                0595  1050          .IF     GT,CA$_MEASURE
                     0014'CF  D6  0595  1051          INCL    W^PMS$GL_DZROFLTS           ;COUNT DEMAND ZERO PAGE FAULTS
                                0599  1052          .ENDC
                                0599  1053
                                0599  1054 ;
                                0599  1055 ; ***** BE AWARE THAT THE FOLLOWING CLRB ASSUMES THAT BIT 8 IS NOT
                                0599  1056 ; ***** IN USE FOR ANY OF THE PAGE TYPE FLAGS, ETC.
                                0599  1057 ;
                        52  94  0599  1058          CLRB    R2                          ;CLEAR OUT THE PAGE TYPE
 62  0200 8F  00    64  00  2C  059B  1059          MOVC5   #0,(R4),#0,#^X200,(R2)      ;ZERO THE PAGE, PCB ADDRESS TO R1
                        54  51  D0  05A3  1060          MOVL    R1,R4                       ;RECOVER PCB ADDRESS
                                05A6  1061          SETIPL  #IPL$_SYNCH                 ;NOT SWAPPABLE WHILE COMPLETING THE FAULT
                     12  08  DA  05A6                MTPR    #IPL$_SYNCH,S^#PR$_IPL
           53    8E    6C  A4  C1  05A9  1062          ADDL3   PCB$L_PHD(R4),(SP)+,R3      ;RE-BIAS BYTE INDEX TO PTE TO GET SVAPTE
     50    63  7BE00000  8F  CB  05AE  1063          BICL3   #^C<PTE$M_VALID ! PTE$M_MODIFY ! -
                                05B6  1064                  PTE$M_PFN>,(R3),R0          ;EVEN PFN MIGHT HAVE CHANGED
              63    8E    50  C9  05B6  1065          BISL3   R0,(SP)+,(R3)               ;SET PTE WITH CORRECT PROT AND OWNER
                                05BA  1066          INVALID 8(SP),R2                    ;INVALIDATE TRANSLATION BUFFER
                 52    08  AE  D0  05BA                MOVL    8(SP),R2
                     3A  52  DA  05BE                MTPR    R2,S^#PR$_TBIS
                     51  6E  7D  05C1  1067          MOVQ    (SP),R1                     ;SEE IF DZRO GLOBAL
                         3D  13  05C4  1068          BEQL    60$                         ;BRANCH IF NOT, BOTH ARE ZERO
                     62  51  D1  05C6  1069          CMPL    R1,(R2)                     ;STILL THE SAME PTE CONTENT?
                         2D  13  05C9  1070          BEQL    40$                         ;BRANCH IF YES, NO RACE TO ZERO THE PAGE
                                05CB  1071 ;
                                05CB  1072 ; MORE THAN ONE PROCESS STARTED TO ZERO THIS GLOBAL DEMAND ZERO PAGE
                                05CB  1073 ; THIS PROCESS LOST THE RACE, CLEAN UP AND REFAULT.
                                05CB  1074 ;
     50  50   15    00  EF  05CB  1075          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R0,R0 ;ISOLATE THE PFN
            6E    50  D0  05D0  1076          MOVL    R0,(SP)                     ;AND SAVE IT
          55    6C  A4  D0  05D3  1077          MOVL    PCB$L_PHD(R4),R5            ;GET PROCESS HEADER ADDRESS
           0000'DF40  B6  05D7  1078          INCW    @W^PFN$AW_REFCNT[R0]        ;RELEASE WSLE AND SHRCNT, BUT NOT PFN
          51    10  A5  3C  05DC  1079          MOVZWL  PHD$W_WSNEXT(R5),R1         ;WORKING SET LIST INDEX
          52    08  AE  D0  05E0  1080          MOVL    8(SP),R2                    ;VIRTUAL ADDRESS (LOW BITS = PAGTYP)
                032E  30  05E4  1081          BSBW    MMG$FREWSLX                 ;FREE THE WORKING SET LIST ENTRY
                04 50   E8  05E7  1082          BLBS    R0,30$                      ;BRANCH IF SUCCESSFUL
                                05EA  1083 ;
                                05EA  1084 ; FREWSLX COULD ONLY RETURN FAILURES STATUS IF PAGE FILE BACKING STORE
                                05EA  1085 ; NEEDED TO BE RESERVED AND THERE WAS NONE AVAILABLE.  THIS PAGE ALREADY
                                05EA  1086 ; HAS BACKING STORE, SO THIS CANNOT HAPPEN.
                                05EA  1087 ;
                                05EA  1088          BUG_CHECK FREWSLX,FATAL
                         FEFF   05EA                        .WORD   ^XFEFF
                         0004'  05EC                        .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_FREWSLX!4
                                05EE  1089
                     09  BA  05EE  1090 30$:    POPR    #^M<R0,R3>                  ;GET PFN AND MASTER PTE ADDRESS
                   FA0D'  30  05F0  1091          BSBW    MMG$RLPFNSAVPTE             ;RELEASE PFN, SAVE PTE CONTENTS
              5E    08  C0  05F3  1092          ADDL    #8,SP                       ;CLEAN OFF VA, PROCESS PTE
                    0E  11  05F6  1093          BRB     PGFCOMPLETE
                                05F8  1094 ;
                                05F8  1095 ; GLOBAL DEMAND ZERO PAGE, MAKE MASTER PTE VALID TOO
                                05F8  1096 ;
  51    867FFFFF  8F  CA  05F8  1097 40$:    BICL    #^C<PTE$M_PROT ! PTE$M_OWN>,R1 ;MASTER PTE PROTECTION AND OWNER
```

```
                62 51   50   C9  05FF  1098          BISL3   R0,R1,(R2)              ;SET MASTER PTE VALID
                   5E   10   C0  0603  1099 60$:      ADDL    #4*4,SP                 ;CLEAN OFF 4 LONG WORDS
                                 0606  1100          .dsabl  lsb
                                 0606  1101 PGFCOMPLETE:
                        3F   BA  0606  1102          POPR    #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE THE REGISTERS
                   5E   08   C0  0608  1103          ADDL    #8,SP                   ;CLEAN OFF THE EXCEPTION PARAMETERS
                             02  0608  1104          REI                             ;AND RETURN FROM THE EXCEPTION
```

```
                                060C  1106          .SBTTL  FREE, MODIFIED, OR BAD PAGE LIST, RELEASE PENDING
                                060C  1107 ;
                                060C  1108 ; THIS IS A FAULT OFF THE FREE, MODIFIED, OR BAD PAGE LISTS
                                060C  1109 ; R0 = PFN, R2 = LISTID,
                                060C  1110 ; 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE
                                060C  1111 ;
                                060C  1112          .ENABL  LSB
                                060C  1113 PFNLIST:
                    F9F1'   30  060C  1114          BSBW    MMG$REMPFN              ;REMOVE PFN FROM LIST
                                060F  1115 WRITEINPROG:
                                060F  1116 RELEASEPEND:
                52    6E    7D  060F  1117          MOVQ    (SP),R2                 ;R2 = VA, R3 = SVAPTE
                    047C    30  0612  1118          BSBW    MMG$MAKEWSLE            ;MAKE A WORKING SET LIST ENTRY
                52    8E    7D  0615  1119          MOVQ    (SP)+,R2                ;R2=VA, R3=SVAPTE
                                0618  1120 ;
                                0618  1121 ; SET PAGE ACTIVE AND VALID
                                0618  1122 ;
          51    0000'DF40     D0  0618  1123          MOVL    @W^PFN$AL_PTE[R0],R1    ;GET MASTER PTE ADDRESS
                                061E  1124
                                061E  1125          ASSUME  PFN$V_DELCON EQ PFN$V_LOC+PFN$S_LOC+1 ;DELCON IS 2ND BIT TO LEFT OF
                                061E  1126                                          ;BIT IN BETWEEN IS FOR LOC EXPANSION
             05    00    07   F0  061E  1127          INSV    #PFN$C_ACTIVE,#PFN$V_LOC,#PFN$S_LOC+2,- ;SET PAGE ACTIVE
                0000'DF40       0622  1128                  @W^PFN$AB_STATE[R0]      ;AND CLEAR DELCON
             00 61    1F   E2  0626  1129          BBSS    #PTE$V_VALID,(R1),50$    ;SET VALID BIT
                04    52   91  062A  1130 50$:     CMPB    R2,#WSL$C_GLOBAL        ;GLOBAL OR PAGE TABLE PAGE?
                      06   18  062D  1131          BGEQ    100$                    ;BRANCH IF YES
                      3F   BA  062F  1132 60$:     POPR    #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE SAVED REGISTERS
                5E    08   C0  0631  1133          ADDL    #8,SP                   ;CLEAN OFF THE EXCEPTION PARAMETERS
                           02  0634  1134          REI                             ;AND RETURN FROM FAULT
                                0635  1135 ;
                                0635  1136 ; GLOBAL PAGE OR PAGE TABLE PAGE
                                0635  1137 ;
                      08    52  91  0635  1138 100$:    CMPB    R2,#WSL$C_PPGTBL        ;PROCESS PAGE TABLE PAGE?
                            16  18  0638  1139          BGEQ    120$                    ;BRANCH IF PROCESS OR GLOBAL PAGE TABLE
                                063A  1140 ;
                                063A  1141 ; GLOBAL PAGE
                                063A  1142 ;
          51    61    7BE00000  8F  CB  063A  1143          BICL3   #^C<PTE$M_VALID ! PTE$M_MODIFY ! PTE$M_PFN>,(R1),R1 ;MASTER PTE
                                0642  1144 ;
                                0642  1145 ; R1 = VALID AND PFN BITS TO STORE INTO SLAVE PTE
                                0642  1146 ; R3 = SLAVE PTE ADDRESS
                                0642  1147 ;
                                0642  1148 SETSLAVEPTE:
          52    63    867FFFFF  8F  CB  0642  1149          BICL3   #^C<PTE$M_PROT ! PTE$M_OWN>,(R3),R2 ;PROTECTION AND OWNER FROM SLAVE
                   63    52   51  C9  064A  1150          BISL3   R1,R2,(R3)              ;STORE THE NEW SLAVE PTE
                         DF   11  064E  1151          BRB     60$                     ;AND EXIT THROUGH COMMON CODE
                                0650  1152 ;
                                0650  1153 ; PROCESS OR GLOBAL PAGE TABLE
                                0650  1154 ;
                         DD   14  0650  1155 120$:    BGTR    60$                     ;BRANCH IF GLOBAL PAGE TABLE
                         51   DD  0652  1156          PUSHL   R1                      ;SAVE REGISTER AROUND THE CALL
                      0549    30  0654  1157          BSBW    MMG$DECPHDREF           ;ONE LESS LOCK ON HDR SPTE
                         02   BA  0657  1158          POPR    #^M<R1>                 ;RESTOR REGISTER
                         D4   11  0659  1159          BRB     60$                     ;AND EXIT THROUGH COMMON CODE
                                065B  1160
                                065B  1161          .DSABL  LSB
                                065B  1162 ;
```

```
                                065B  1163 ; FAILED TO READ THE DESIRED PAGE, RELEASE THE PFN AND ISSUE AN EXCEPTION
                                065B  1164 ; R0     = PAGE FRAME NUMBER
                                065B  1165 ; R4     = PROCESS CONTROL BLOCK ADDRESS
                                065B  1166 ; R5     = SYSTEM ADDRESS OF PROCESS HEADER
                                065B  1167 ; 0(SP) = VIRTUAL ADDRESS (LOW BITS = PAGE TYPE)
                                065B  1168 ; 4(SP) = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                                065B  1169 ;
                                065B  1170 READERR:
                          0C    BA  065B  1171          POPR    #^M<R2,R3>              ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE
                                065D  1172          PFN_REFERENCE   -
                                065D  1173          MOVZWL  <@W^PFN$Ax_WSLX[R0],R1>,-       ;GET WORKING SET LIST INDEX IF NOT G
                                065D  1174                  LONG_OPCODE=MOVL,-
                                065D  1175                  IMAGE=SYS_NONPAGED
                                065D                .SAVE   LOCAL_BLOCK
                                065D                .PSECT  $ABS$,ABS
                          00000000  004C            .=0
                                065D                .RESTORE
                                065D                .SAVE_PSECT     LOCAL_BLOCK
                                0000                .PSECT  Z$INIT$PFN_FIXUP_TABLE
                          0000065D' 0000            .ADDRESS        ..PFN
                                3C  0004            .BYTE   OP$_MOVZWL
                                D0  0005            .BYTE   OP$_MOVL
                                065D                .RESTORE_PSECT
                51    0000'DF40   3C  065D          MOVZWL  @W^PFN$Ax_WSLX[R0],R1
                                0663  1176
                                0663  1177          ASSUME  PFN$C_PROCESS EQ 0
                                0663  1178          ASSUME  PFN$C_SYSTEM  EQ 1
                                0663  1179          ASSUME  PFN$C_GLOBAL  EQ 2
                                0663  1180          ASSUME  PFN$C_GBLWRT  EQ 3
                                0663  1181          ASSUME  PFN$C_PPGTBL  EQ 4
                                0663  1182          ASSUME  PFN$C_GPGTBL  EQ 5
          02    52    03    01  EC  0663  1183          CMPV    #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,#PFN$C_GLOBAL ;IS PAGE GLOBAL?
                      18    19  0668  1184          BLSS    20$                     ;BRANCH IF NOT
                                066A  1185 ;
                                066A  1186 ; GLOBAL PAGE DOES NOT HAVE WORKING SET LIST INDEX IN WSLX ARRAY
                                066A  1187 ; MUST SCAN THE PROCESS' WORKING SET LIST FOR THE VIRTUAL ADDRESS
                                066A  1188 ;
                      53    55  C2  066A  1189          SUBL    R5,R3                   ;UNBIAS SVAPTE, COULD BE SWAPPED HERE
          55    00000000'9F     D0  066D  1190          MOVL    @#CTL$GL_PHD,R5         ;USE P1 SPACE HEADER WINDOW
                      F989'     30  0674  1191          BSBW    MMG$SCNWSLX             ;SCAN FOR THE WORKING SET LIST INDEX
                                0677  1192 ;
                                0677  1193 ; COULD HAVE BEEN SWAPPED IN THE ABOVE ROUTINE, BUT IPL IS BACK AT SYNCH NOW
                                0677  1194 ;
                      55  6C A4  D0  0677  1195          MOVL    PCB$L_PHD(R4),R5        ;RECOVER SYSTEM ADDRESS OF PHD
                          53    55  C0  067B  1196          ADDL    R5,R3                   ;REBIAS PTE ADDRESS
                                51  D5  067E  1197          TSTL    R1                      ;SEE IF FOUND WORKING SET LIST ENTRY
                                3E  13  0680  1198          BEQL    40$                     ;IF NOT, PAGE WENT AWAY OR SOME OTHER
                                        0682  1199                                          ;PROCESS WAS THE ORIGINATOR OF THE I/O
                       52   6541  D0  0682  1200 20$:      MOVL    (R5)[R1],R2             ;FETCH WORKING SET LIST ENTRY
                              0C  BB  0686  1201          PUSHR   #^M<R2,R3>              ;SAVE VIRTUAL ADDRESS AND PTE ADDRESS
                            028A  30  0688  1202          BSBW    MMG$FREWSLX             ;FREE THIS WORKING SET LIST ENTRY
                              0C  BA  068B  1203          POPR    #^M<R2,R3>              ;RECOVER VA AND SVAPTE
                          04  50  E8  068D  1204          BLBS    R0,30$                  ;BRANCH IF SUCCESSFUL
                                0690  1205 ;
                                0690  1206 ; FREWSLX CAN ONLY FAIL IF PAGE FILE NEEDED TO BE ALLOCATED AND IT COULDN'T BE
                                0690  1207 ; THIS CASE IS NOT POSSIBLE HERE.
                                0690  1208 ;
```

```
                                0690  1209          BUG_CHECK FREWSLX,FATAL
                        FEFF    0690                        .WORD   ^XFEFF
                        0004'   0692                        .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_FREWSLX!4
                                0694  1210 ;
                                0694  1211 ; IF THIS PAGE FAULT IS FROM USER OR SUPER MODE THEN ISSUE A
                                0694  1212 ; PAGE READ ERROR EXCEPTION.
                                0694  1213 ;
                    02  18  ED   0694  1214 30$:    CMPZV   #PSL$V_CURMOD,#PSL$S_CURMOD,- ;IF FAULTING MODE IS
                  02    24 AE   0697  1215                  FLTPSL(SP),#PSL$C_SUPER ;USER OR SUPER
                        24  18  069A  1216          BGEQ    40$                     ;THEN PAGE READ ERROR EXCEPTION
                                069C  1217 ;
                                069C  1218 ; THIS IS A BAD SITUATION NOW, AN EXCEPTION IN EXEC OR KERNEL MODE WILL
                                069C  1219 ; CRASH THE SYSTEM.  IF THIS PAGE IS OWNED BY USER OR SUPER THEN TRY
                                069C  1220 ; SUBSTITUTING A PAGE OF ZEROS.  THIS SHOULD SATISFY THE SYSTEM CODE WHICH
                                069C  1221 ; IS ACCESSING THE PROCESS PAGE SINCE IT IS PARANOID ABOUT USER SUPPLIED
                                069C  1222 ; DATA.  THE NEW PAGE WILL BE EXEC READ WRITE BUT OWNED BY THE ORIGINAL
                                069C  1223 ; OWNER.  THIS WILL RESULT IN AN ACCESS VIOLATION WHEN THE PAGE IS TOUCHED
                                069C  1224 ; IN USER OR SUPER MODE.
                                069C  1225 ;
          50    63  02  17  EF  069C  1226          EXTZV   #PTE$V_OWN,#PTE$S_OWN,(R3),R0 ;GET THE PAGE OWNER
                    02  50  D1  06A1  1227          CMPL    R0,#PSL$C_SUPER         ;OWNED BY USER OR SUPER?
                        1A  19  06A4  1228          BLSS    40$                     ;BRANCH IF NOT, READ ERROR FOR
                                06A6  1229                                          ;A CRUCIAL PAGE, ISSUE THE PAGE
                                06A6  1230                                          ;READ ERROR EXCEPTION, DOWN WE GO.
                                06A6  1231          SETIPL  #IPL$_ASTDEL            ;LOWEST POSSIBLE FAULT IPL
                    12  02  DA  06A6                      MTPR    #IPL$_ASTDEL,S^#PR$_IPL
                                06A9  1232 ;
                                06A9  1233 ; FORM ARGUMENT LIST FOR CRETVA
                                06A9  1234 ;
                        52  DD  06A9  1235          PUSHL   R2                      ;VIRTUAL ADDRESS TO CREATE
                        05  BB  06AB  1236          PUSHR   #^M<R0,R2>              ;ANOTHER COPY OF ADR TO FORM RANGE
                                06AD  1237                                          ;ACCESS MODE PARAMETER
                01 AE   05  90  06AD  1238          MOVB    S^#PRT$C_EW,1(SP)       ;SET DESIRED PAGE PROTECTION
                        00  DD  06B1  1239          PUSHL   #0                      ;NULL RETURN ADDRESS
                    08  AE  DF   06B3  1240          PUSHAL  8(SP)                   ;ADDRESS OF RANGE TO CREATE
00000000'GF             05  FB  06B6  1241          CALLS   #5,G^MMG$CRETVA         ;KERNEL MODE ENTRY TO CRETVA
                                06BD  1242                                          ;PRESERVES IPL
                                06BD  1243                                          ;STRIP OFF INPUT RANGE WHEN DONE
                      FF46  31  06BD  1244          BRW     PGFCOMPLETE             ;FAULT THIS PAGE FROM SCRATCH
                    1A  AE  B6  06C0  1245 40$:     INCW    FLTCTL+2(SP)            ;INDICATE PAGE READ ERROR
                                06C3  1246 ACVIOLAT:
                50  26  AE  3C  06C3  1247          MOVZWL  FLTPSL+2(SP),R0         ;GET IPL FROM FAULT PSL
                                06C7  1248          ENBINT  R0                      ;AND RESTORE IT
                    12  50  DA  06C7                      MTPR    R0,S^#PR$_IPL
                        3F  BA  06CA  1249          POPR    #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE REGISTERS SAVED BY PAGE FAULT
                03  6E  10  E5  06CC  1250          BBCC    #16,(SP),10$            ;BRANCH IF ACCESS VIOLATION
                      F92D' 31  06D0  1251          BRW     EXE$PAGRDERR            ;ISSUE THE EXCEPTION
                      F92A' 31  06D3  1252 10$:     BRW     EXE$ACVIOLAT            ;ACCESS VIOATION
```

```
                                06D6  1254          .SBTTL  SCANDEADPT - SCAN A DEAD PAGE TABLE FOR TRANSITION PAGES
                                06D6  1255 ;
                                06D6  1256 ; INPUTS:
                                06D6  1257 ;
                                06D6  1258 ;       R2 = VIRTUAL ADDRESS OF PAGE TABLE (LOW BITS = PAGE TYPE)
                                06D6  1259 ;       R5 = PROCESS HEADER ADDRESS
                                06D6  1260 ;       IPL = SYNCH
                                06D6  1261 ;
                                06D6  1262 ; OUTPUTS:
                                06D6  1263 ;
                                06D6  1264 ;       NONE
                                06D6  1265 ;
                                06D6  1266 SCANDEADPT:
     50    70 A5    6E A5    A3  06D6  1267          SUBW3   PHD$W_PTCNTVAL(R5),PHD$W_PTCNTACT(R5),R0 ;ACTIVE PAGE TABLES
                                06DC  1268                                                          ;THAT DON'T CONTAIN VALID WSLE'S
                                06DC  1269                                                          ;ARE 'DEAD PAGE TABLES'
                      03    14  06DC  1270          BGTR    10$
                    0085    31  06DE  1271          BRW     50$                                     ;BRANCH IF NO DEAD PAGE TABLES
          50    6C A5    A0  06E1  1272 10$:      ADDW    PHD$W_PTCNTLCK(R5),R0                   ;ADD IN THE LOCKED PAGE TABLES
          50    74 A5    A0  06E5  1273          ADDW    PHD$W_WSFLUID(R5),R0                    ;NEED TWICE FLUID EXTRA
          50    74 A5    A0  06E9  1274          ADDW    PHD$W_WSFLUID(R5),R0
             50    50    3C  06ED  1275          MOVZWL  R0,R0                                   ;GET IT IN A LONGWORD
          7E    50 A5    3C  06F0  1276          MOVZWL  PHD$W_WSSIZE(R5),-(SP)                  ;GET CURRENT WS SIZE
             50    6E    C2  06F4  1277          SUBL    (SP),R0                                 ;SUBTRACT OUT CURRENT WSL SIZE
     6E    0E A5    08 A5    A3  06F7  1278          SUBW3   PHD$W_WSLIST(R5),PHD$W_WSDYN(R5),(SP) ;GET LOCKED PORTION OF WS
             6E    6E    3C  06FD  1279          MOVZWL  (SP),(SP)                               ;GET IT IN A LONGWORD
             50    8E    C0  0700  1280          ADDL    (SP)+,R0                                ;ADD IN LOCKED ENTRY COUNT
                   61    19  0703  1281          BLSS    50$                                     ;BRANCH IF SAFE TO POSTPONE DEAD PAGE
                               0705  1282                                                          ;TABLE SCAN
      50    52    00C8 C5    C3  0705  1283          SUBL3   PHD$L_POBR(R5),R2,R0                    ;BYTE OFFSET FROM FIRST PO PAGE TABLE
                   59    19  0708  1284          BLSS    50$                                     ;BRANCH IF PROCESS HEADER PAGE
        50    50    F7 8F    78  070D  1285          ASHL    #-9,R0,R0                               ;FORM PAGE NUMBER
             50    55    C0  0712  1286          ADDL    R5,R0                                   ;ADD IN PHD BASE
          50    68 A5    C0  0715  1287          ADDL    PHD$L_PTWSLEVAL(R5),R0                  ;ADD IN OFFSET TO BYTE ARRAY OF COUNTS
                               0719  1288                                                          ;OF VALID WSLE'S IN EACH PAGE TABLE
                   60    95  0719  1289          TSTB    (R0)                                    ;IS THIS A DEAD PAGE TABLE
                   49    18  071B  1290          BGEQ    50$                                     ;BRANCH IF NOT
                               071D  1291 ;
                               071D  1292 ; R1,R2,R3 ARE PRESERVED UP TO THIS POINT
                               071D  1293 ;
          10 A5    51    B0  071D  1294          MOVW    R1,PHD$W_WSNEXT(R5)                     ;UPDATE NEXT POINTER
            0038'CF    D6  0721  1295          INCL    W^PMS$GL_DPTSCN                         ;COUNT THESE SCANS
     50    52    15    09  EF  0725  1296          EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R0 ;PAGE NUMBER OF THE PT IN SYSTEM SPACE
        7FE00000 8F    CB  072A  1297          BICL3   #^C<PTE$M_VALID ! PTE$M_PFN>,- ;GET PFN AND VALID BIT
        50    0000'DF40       0730  1298                  @W^MMG$GL_SPTBASE[R0],R0                ;FROM SPT ENTRY FOR THE PAGE TABLE
          29 50    1F    E5  0735  1299          BBCC    #PTE$V_VALID,R0,40$                     ;CLEAR VALID BRANCH IF IT WAS CLEAR
                   00    DD  0739  1300          PUSHL   #0                                      ;FLAG FOR MODIFIED PAGE WRITER NEEDED
                               073B  1301                  PFN_REFERENCE   -
                               073B  1302          MOVZWL  <@W^PFN$Ax_SHRCNT[R0],-(SP) >,- ;NUMBER OF TRANSITION PAGES
                               073B  1303                  LONG_OPCODE=MOVL,-
                               073B  1304                  IMAGE=SYS_NONPAGED
                               073B                .SAVE_PSECT     LOCAL_BLOCK
                           00000006                .PSECT  Z$INIT$PFN_FIXUP_TABLE
                  0000073B' 0006                   .ADDRESS                ...PFN
                         3C 000A                   .BYTE   OP$_MOVZWL
                         D0 000B                   .BYTE   OP$_MOVL
                           0000073B                .RESTORE_PSECT
```

```
          7E   0000'DF40   3C  073B              MOVZWL  @W^PFN$Ax_SHRCNT[R0],-(SP)
                      1F   13  0741  1305        BEQL    40$                        ;IF NONE, INCONSISTENT
           52    01FF 8F   AA  0743  1306        BICW    #VA$M_BYTE,R2              ;START SCANNING PT AT BEGINNING
              53    80 8F  9A  0748  1307        MOVZBL  #128,R3                    ;AT MOST 128 PTE'S
   50    82  7BA00000 8F   CB  074C  1308 20$:   BICL3   #^C<PTE$M_VALID ! -        ;GET THE VALID BIT
                               0754  1309                PTE$M_TYP1 ! PTE$M_TYP0 ! - ;PTE TYPE BITS
                               0754  1310                PTE$M_PFN>,(R2)+,R0        ;AND THE PFN FROM THE PTE
                      07   13  0754  1311        BEQL    30$                        ;BRANCH IF DEMAND ZERO PAGE
      51    50    EA 8F    78  0756  1312        ASHL    #-PTE$V_TYP0,R0,R1         ;VALID, TYP1, TYP0 ALL 0 IF TRANSITION
                      0D   13  075B  1313        BEQL    60$                        ;BRANCH IF TRANSITION PAGE
                   EC 53   F5  075D  1314 30$:   SOBGTR  R3,20$                     ;LOOP THROUGH THE PAGE TABLE
                      36   11  0760  1315        BRB     100$                       ;ALL DONE, CNT=# I/O REQ OUTSTANDING
                               0762  1316 40$:   BUG_CHECK SCANDEADPT,FATAL         ;SPT ENTRY FOR PAGE TABLE NOT VALID
                     FEFF.     0762                      .WORD   ^XFEFF
                     0004'     0764                      .IIF IDN <FATAL>,<FATAL> , .WORD      BUG$_SCANDEADPT!4
                               0766  1317
              50    01     D0  0766  1318 50$:   MOVL    #1,R0                      ;SET CONTINUE RATHER THAN RESTART
                           05  0769  1319        RSB
                               076A  1320                                           ;SHRCNT FOR PAGE TABLE IS 0
                               076A  1321                                           ;DIDN'T FIND SHRCNT TRANSITION PAGES
                               076A  1322                                           ;BEFORE RUNNING OFF THE END OF THE PT
                               076A  1323 ;
                               076A  1324 ; THIS IS A TRANSITION PAGE
                               076A  1325 ;
                      0C   BB  076A  1326 60$:   PUSHR   #^M<R2,R3>                 ;SAVE THESE REGISTERS
              03    00     EF  076C  1327        EXTZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,- ;GET PAGE LOCATION
        52   0000'DF40         076F  1328                @W^PFN$AB_STATE[R0],R2     ;FROM THE STATE BYTE
              01    52     D1  0774  1329        CMPL    R2,#PFN$C_MFYPAGLST        ;ON MODIFIED OR FREE PAGE LIST
                      1A   14  0777  1330        BGTR    90$                        ;BRANCH IF NOT ON EITHER
                      0E   13  0779  1331        BEQL    80$                        ;BRANCH IF ON MODIFIED PAGE LIST
                               077B  1332 ;
                               077B  1333 ; PAGE IS ON THE FREE PAGE LIST
                               077B  1334 ;
                   F882'   30  077B  1335        BSBW    MMG$REMPFN                 ;ON FREE LIST, REMOVE IT
        0000'DF40   10     88  077E  1336        BISB    #PFN$M_DELCON,@W^PFN$AB_STATE[R0] ;FORCE DELETE CONTENTS
                   F879'   30  0784  1337        BSBW    MMG$RELPFN                 ;AND RELEASE THE PAGE
                      0A   11  0787  1338        BRB     90$
                               0789  1339 ;
                               0789  1340 ; PAGE IS ON MODIFIED PAGE LIST
                               0789  1341 ;
        0000'DF40   10     88  0789  1342 80$:   BISB    #PFN$M_DELCON,@W^PFN$AB_STATE[R0] ;DELETE CONTENTS AFTER WRITING
            0C AE   01     88  078F  1343        BISB    #1,12(SP)                  ;FLAG MODIFIED PAGE RELEASE NEEDED
                      0C   BA  0793  1344 90$:   POPR    #^M<R2,R3>                 ;RESTORE SAVE REGISTERS
                   C5 6E   F5  0795  1345        SOBGTR  (SP),30$                   ;COUNT DOWN THE TRANSITION COUNT
                      03   BA  0798  1346 100$:  POPR    #^M<R0,R1>                 ;CLEAN OFF THE EXHAUSTED COUNT AND FLAG
                   0E 51   E9  079A  1347        BLBC    R1,110$                    ;BRANCH IF NO MODIFIED PAGE WRITING
                  0000'CF  D4  079D  1348        CLRL    W^SCH$GL_MFYLOLIM          ;MAKE SURE IT'S WRITTEN SOON
                  0000'CF  B4  07A1  1349        CLRW    W^SCH$GL_MFYLIM            ;CLEAR COUNT, NOT WRITE REQUESTED FLAG
              51    0B     3C  07A5  1350        MOVZWL  #RSN$_MPLEMPTY,R1          ;SET RESOURCE TO WAIT FOR
              5E    04     C0  07A8  1351        ADDL    #4,SP                      ;RETURN TO ORIGINAL CALLER
                      50   D4  07AB  1352 110$:  CLRL    R0                         ;SET FAILURE (OR RESTART)
                           05  07AD  1353        RSB
```

```
                                07AE  1355          .SBTTL  WSLEPFN - FETCH PFN FROM WORKING SET LIST ENTRY
                                07AE  1356 ;
                                07AE  1357 ; CALLING SEQUENCE:
                                07AE  1358 ;
                                07AE  1359 ;       BSBW    MMG$WSLEPFN
                                07AE  1360 ;
                                07AE  1361 ; INPUTS:
                                07AE  1362 ;
                                07AE  1363 ;       R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                                07AE  1364 ;            FOR A PAGE THAT IS IN THE WORKING SET LIST
                                07AE  1365 ;
                                07AE  1366 ; OUTPUTS:
                                07AE  1367 ;
                                07AE  1368 ;       R0 = PFN
                                07AE  1369 ;       R2,R3 PRESERVED
                                07AE  1370 ;
                                07AE  1371 WSLEPFNMSK:
                    7B800000    07AE  1372          .LONG   ^C<PTE$M_VALID ! PTE$M_TYP0 ! PTE$M_TYP1 ! PTE$M_PGFLVB>
                                07B2  1373
                                07B2  1374          .ENABL  LSB
                                07B2  1375
                                07B2  1376 MMG$WSLEPFN::
      50    63    F9 AF    CB  07B2  1377          BICL3   B^WSLEPFNMSK,(R3),R0    ;GET VALID, TYP0, TYP1, PFN/GPTX
      51    50    EA 8F    78  07B7  1378          ASHL    #-PTE$V_TYP0,R0,R1      ;SEE IF TRANSITION OR VALID PAGE
                        06    14  07BC  1379          BGTR    FRE_GBLTRANS            ;BRANCH IF NEITHER
   50    50    15    00    EF  07BE  1380 10$:     EXTZV   #PTE$V_PFN,#PTE$S_PFN,R0,R0 ;GET PFN
                        05    05  07C3  1381 20$:     RSB                             ;AND RETURN
                                07C4  1382 ;
                                07C4  1383 ; INPUTS:
                                07C4  1384 ;
                                07C4  1385 ;       R0 = PAGE TABLE ENTRY WITH TYP0, TYP1 AND GPTX BITS
                                07C4  1386 ;       R1 = RESULT OF SHIFTING R0 BY -PTE$V_TYP0
                                07C4  1387 ;          = 1 IF PAGE IS GLOBAL
                                07C4  1388 ;          > 1 IF PAGE TYPE IS INVALID FOR THIS CONTEXT
                                07C4  1389 ;
                                07C4  1390 ; OUTPUTS:
                                07C4  1391 ;
                                07C4  1392 ;       R0 = PFN IF GLOBAL PAGE IN TRANSITION
                                07C4  1393 ;       BUGCHK IF NOT
                                07C4  1394 ;
                                07C4  1395 FRE_GBLTRANS:
                  13 51    F5  07C4  1396          SOBGTR  R1,WSLVANVAL            ;BRANCH IF NOT GLOBAL FORMAT
            0F 50    16    E5  07C7  1397          BBCC    #PTE$V_TYP0,R0,WSLVANVAL ;CLEAR TYP0, MUST HAVE BEEN SET
50  0000'DF40    E0 AF    CB  07CB  1398          BICL3   B^WSLEPFNMSK,@W^MMG$GL_GPTBASE[R0],R0 ;FETCH MASTER PTE
      51    50    EA 8F    78  07D3  1399          ASHL    #-PTE$V_TYP0,R0,R1      ;MAKE SURE THIS IS IN TRANSITION
                        E9    13  07D8  1400          BEQL    20$                     ;BRANCH IF IT IS, R0 = PFN
                                07DA  1401 WSLVANVAL:
                                07DA  1402          BUG_CHECK WSLVANVAL,FATAL       ;WORKING SET LIST ENTRY VIRTUAL
                          FEFF  07DA                        .WORD   ^XFEFF
                          0004' 07DC                        .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_WSLVANVAL!4
                                07DE  1403                                          ;ADDRESS IS NOT VALID
                                07DE  1404          .DSABL  LSB
```

```
                        .SBTTL  FREWSLE - FREE A WORKING SET LIST ENTRY
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE CHOOSES A WORKING SET LIST ENTRY, RELEASES THE
                ; PAGE WHICH OCCUPIES IT (IF ANY), MARKS THE ENTRY AVAILABLE, AND
                ; LEAVES THE WSNEXT POINTING TO THE AVAILABLE ENTRY.
                ;       IN RELEASING A PAGE, IF ITS BACKING STORE ADDRESS IS A
                ; "NOT YET ALLOCATED" PAGING FILE ADDRESS, THEN A PAGING FILE VBN
                ; IS ALLOCATED AT THIS TIME.  IT IS POSSIBLE THAT NO VBN'S ARE AVAILABLE
                ; AND THUS THIS ROUTINE CAN RETURN UNSUCCESSFULLY.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW    MMG$FREWSLE
                ;
                ; INPUT PARAMETERS:
                ;
                ;       R4 = PCB ADDRESS
                ;       R5 = PROCESS HEADER ADDRESS - MAY BE P1 SPACE ADDRESS
                ;               IF WORKING WITH PROCESS WORKING SET LIST
                ;       IPL = SYNCH
                ;
                ; IMPLICIT INPUTS:
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       IF SUCCESSFUL
                ;               R0 LOW BIT IS SET
                ;       IF NOT SUCCESSFUL
                ;               R0 LOW BIT IS CLEAR AND
                ;               R1 = RESOURCE TO WAIT FOR (#RSN$_XXXXX)
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       IF A WORKING SET ENTRY WAS FREED,  IT IS PLACED ON THE FREE LIST
                ;
                ; COMPLETION CODES:
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;       NONE
                ;
                ;--
```

```
                                07DE  1452 ; FOR MMG$FREWSLX ENTRY POINT
                                07DE  1453 ;
                                07DE  1454 ; INPUTS:
                                07DE  1455 ;
                                07DE  1456 ;       R1  = WORKING SET LIST INDEX
                                07DE  1457 ;       R2  = VIRTUAL ADDRESS (LOW BITS  = PAGE TYPE)
                                07DE  1458 ;       R3  = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                                07DE  1459 ;       R4  = PROCESS CONTROL BLOCK ADDRESS
                                07DE  1460 ;       R5  = PROCESS HEADER ADDRESS
                                07DE  1461 ;       IPL = SYNCH
                                07DE  1462 ;
                                07DE  1463 ; OUTPUTS:
                                07DE  1464 ;
                                07DE  1465 ;       R0 = STATUS
                                07DE  1466 ;       R1 = RESOURCE TO WAIT FOR IF NOT SUCCESSFUL
                                07DE  1467 ;
                                07DE  1468
                                07DE  1469 ;
                                07DE  1470 ; FOUND AN EMPTY WORKING SET LIST ENTRY, CHECK WHETHER THERE IS A
                                07DE  1471 ; NEW PEAK WORKING SET SIZE AND WHETHER SWAP AREA NEEDS TO GROW.
                                07DE  1472 ;
                                07DE  1473 ; R0 = GPGCNT+PPGCNT
                                07DE  1474 ;
                                07DE  1475         .ENABLE LSB
                                07DE  1476
      11 36 A5   02   E1        07DE  1477 10$:    BBC     #PHD$V_WSPEAKCHK,PHD$W_FLAGS(R5),15$ ;BRANCH IF CANNOT BE
                                07E3  1478                                                      ;ABOVE PREVIOUS PEAK WORKING SET SIZE
00000000'GF   50   B1           07E3  1479         CMPW    R0,G^CTL$GL_WSPEAK                  ;ABOVE PREVIOUS RECORDED PEAK?
                   08   1F      07EA  1480         BLSSU   15$                                 ;BRANCH IF NOT
00000000'GF   50   01   A1      07EC  1481         ADDW3   #1,R0,G^CTL$GL_WSPEAK               ;YES, NEW PEAK INCLUDES THE PAGE
                                07F4  1482                                                      ;ABOUT TO BE ADDED TO THE WORKING SET
      11 36 A5   04   E1        07F4  1483 15$:    BBC     #PHD$V_IWSPEAKCK,PHD$W_FLAGS(R5),20$ ;BRANCH IF CANNOT BE
                                07F9  1484                                                      ;ABOVE PREVIOUS PEAK WORKING SET SIZE
00000000'GF   50   B1           07F9  1485         CMPW    R0,G^CTL$GL_IWSPEAK                 ;ABOVE PREVIOUS RECORDED PEAK?
                   08   1F      0800  1486         BLSSU   20$                                 ;BRANCH IF NOT
00000000'GF   50   01   A1      0802  1487         ADDW3   #1,R0,G^CTL$GL_IWSPEAK              ;YES, NEW PEAK INCLUDES THE PAGE
                                080A  1488                                                      ;ABOUT TO BE ADDED TO THE WORKING SET
         10 A5   51   B0        080A  1489 20$:    MOVW    R1,PHD$W_WSNEXT(R5)                 ;UPDATE NEXT POINTER
         52 A5   50   B1        080E  1490         CMPW    R0,PHD$W_SWAPSIZE(R5)               ;IS THERE ENOUGH ROOM TO SWAP PROCESS?
                   2B   1F      0812  1491         BLSSU   30$                                 ;BRANCH IF YES
   50    0A A5    08 A5   A3    0814  1492         SUBW3   PHD$W_WSLIST(R5),PHD$W_WSAUTH(R5),R0 ;GET AUTHORIZED QUOTA
         50   52 A5   B1        081A  1493         CMPW    PHD$W_SWAPSIZE(R5),R0               ;ENOUGH SPACE TO COVER QUOTA?
                   1F   1E      081E  1494         BGEQU   30$                                 ;BRANCH IF SO, DON'T NEED ANY MORE
         50   20 A4   D0        0820  1495         MOVL    PCB$L_WSSWP(R4),R0                  ;GET BLOCK LOCATION OF LAST ALLOCATION
                   19   13      0824  1496         BEQL    30$                                 ;BRANCH IF NON SWAPPING TYPE PROCESS
         51   52 A5   3C        0826  1497         MOVZWL  PHD$W_SWAPSIZE(R5),R1               ;GET CURRENT SIZE
      52   0000'CF   3C        082A  1498         MOVZWL  W^SWP$GW_SWPINC,R2                  ;NEW INCREMENT
              52   51   C0      082F  1499         ADDL    R1,R2                               ;NEW DESIRED SIZE
                F7CB'   30      0832  1500         BSBW    MMG$ALLOCSWPAREA                    ;ALLOCATE A SWAP AREA, R0-R3 CHANGED
                   0E   15      0835  1501         BLEQ    40$                                 ;BRANCH IF ALLOCATION FAILED
         20 A4   50   D0        0837  1502         MOVL    R0,PCB$L_WSSWP(R4)                  ;UPDATE SWAP FILE VBN
         52 A5   52   B0        083B  1503         MOVW    R2,PHD$W_SWAPSIZE(R5)               ;AND SIZE OF AREA
              50   01   3C      083F  1504 30$:    MOVZWL  #SS$_NORMAL,R0                      ;SUCCESSFUL RETURN INDICATION
                        05      0842  1505         RSB
                                0843  1506
                   99   11      0843  1507 35$:    BRB     10$                                 ;GET BRANCH DESTINATION TO REACH
                                0845  1508
```

```
                   53  D4  0845 40$:     CLRL    R3                       ;FORCE SKIP COUNT TO ZERO
          51   10 A5   3C  0847          MOVZWL  PHD$W_WSNEXT(R5),R1      ;INDEX TO NEXT CANDIDATE TO DISCARD
   50  34 A4   36 A4   A1  084B          ADDW3   PCB$W_PPGCNT(R4),PCB$W_GPGCNT(R4),R0 ;CURRENT PAGE COUNT IN USE
   7E  18 A5   08 A5   A3  0851          SUBW3   PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),-(SP) ;GET QUOTA
              8E   50  B1  0857          CMPW    R0,(SP)+                 ;IS PAGE COUNT WITHIN QUOTA OR EXTENT?
                   65  1A  085A          BGTRU   90$                      ;BRANCH IF GREATER THAN QUOTA TO FREE
              0E A4    B5  085C          TSTW    PCB$W_MTXCNT(R4)         ;DOES PROCESS HOLD ANY MUTEX?
                   60  12  085F          BNEQ    90$                      ;BR IF YES, DO NOT LET IT WAIT
                           0861                                           ;A WSLE, I.E. RUN RATHER THAN WAIT
              51   0A  3C  0861          MOVZWL  #RSN$_SWPFILE,R1         ;R1 = RESOURCE TO WAIT FOR
                   50  D4  0864          CLRL    R0                       ;RETURN FAILURE INDICATION
                       05  0866          RSB                              ;RETURN RESOURCE TO WAIT FOR IN R1
                           0867
                           0867 LCKWSLE_NOTPGTB:
                           0867          BUG_CHECK BADLCKWSLE,FATAL       ;LOCKED WORKING SET LIST ENTRY NOT PAGE TABL
                     FEFF  0867                  .WORD   ^XFEFF
                    0004'  0869                  .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_BADLCKWSLE!4
                           086B WSSIZEERR:
                           086B          BUG_CHECK WSSIZEERR,FATAL        ;VALUE OF WSSIZE IS SMALLER THAN IN USE
                     FEFF  086B                  .WORD   ^XFEFF
                    0004'  086D                  .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_WSSIZEERR!4
                           086F WSLENOVAL:
                           086F          BUG_CHECK WSLENOVAL,FATAL        ;WSL ENTRY NOT VALID
                     FEFF  086F                  .WORD   ^XFEFF
                    0004'  0871                  .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_WSLENOVAL!4
                           0873 ;
                           0873 ; THIS IS A LOCKED PAGE IN THE DYNAMIC PORTION OF THE WORKING SET LIST
                           0873 ; IT MUST BE A PAGE TABLE PAGE, NOTE THE CONDITION CODES ARE STILL SET
                           0873 ; FROM THE FETCH OF THE WORKING SET LIST ENTRY.
                           0873 ;
                   F2  18  0873 50$:     BGEQ    LCKWSLE_NOTPGTB          ;BRANCH IF THIS IS NOT A PAGE TABLE
           14 52   01  E0  0875          BBS     #WSL$V_PAGTYP,R2,60$     ;BRANCH TO SKIP GLOBAL PAGE TABLE PAGES
                 FE5A  30  0879          BSBW    SCANDEADPT               ;SCAN THE PAGE TABLE TO SEE IF "DEAD"
                           087C                                           ;IF SO RID IT OF TRANSITION PAGES.
              0E   50  E8  087C          BLBS    R0,60$                   ;AND NOW GET A FREE WS LIST ENTRY
                           087F MMG$FREWSLE::
          53   0000'CF 3C  087F          MOVZWL  W^SGN$GW_WSLMXSKP,R3     ;MAX NUMBER OF TB VALID ENTRIES TO SKIP
          51   10 A5   3C  0884          MOVZWL  PHD$W_WSNEXT(R5),R1      ;INDEX TO NEXT CANDIDATE TO DISCARD
                 6541  D5  0888          TSTL    (R5)[R1]                 ;IS THIS ENTRY FREE?
                   12  13  088B          BEQL    80$                      ;BRANCH IF SO, CHECK FOR TRULY FREE
                   51  D6  088D 60$:     INCL    R1                       ;STEP TO NEXT ENTRY
           12 A5   51  B1  088F          CMPW    R1,PHD$W_WSLAST(R5)      ;AT THE END YET?
                   04  1B  0893          BLEQU   70$                      ;CONTINUE
          51   0E A5   3C  0895          MOVZWL  PHD$W_WSDYN(R5),R1       ;BACK TO THE TOP
           52    6541  D0  0899 70$:     MOVL    (R5)[R1],R2              ;R2 = VA FROM WSLE
                   41  12  089D          BNEQ    120$                     ;BRANCH IF ENTRY IN USE
   50  34 A4   36 A4   A1  089F 80$:     ADDW3   PCB$W_PPGCNT(R4),PCB$W_GPGCNT(R4),R0 ;CURRENT PAGE COUNT IN USE
           50 A5   50  B1  08A5          CMPW    R0,PHD$W_WSSIZE(R5)      ;ARE PAGES IN USE = WORKING SET SIZE?
                   16  13  08A9          BEQL    90$                      ;BRANCH IF SO, NEED TO REPLACE A PAGE
                   BE  1A  08AB          BGTRU   WSSIZEERR                ;BRANCH IF MORE PAGES IN WS THAN IN WSSIZE
   52  18 A5   08 A5   A3  08AD          SUBW3   PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),R2 ;QUOTA NUMBER OF PAGES-1
              52   50  B1  08B3          CMPW    R0,R2                    ;ARE WE WITHIN QUOTA NUMBER OF PAGES?
                   88  1B  08B6          BLEQU   35$                      ;BRANCH IF SO, ALLOWED ANOTHER PAGE
  0000'CF    0000'CF   D1  08B8          CMPL    W^SCH$GL_GROWLIM,W^SCH$GL_FREECNT ;ENOUGH FREE PAGES TO EXTEND?
                   82  19  08BF          BLSS    35$                      ;BRANCH IF SO
      50   51      01  C3  08C1 90$:     SUBL3   #1,R1,R0                 ;SAVE INDEX OF LAST NON-ZERO WSLE
                   51  D6  08C5 100$:    INCL    R1                       ;STEP TO NEXT ENTRY
```

```
                12 A5    51    B1  08C7  1560          CMPW    R1,PHD$W_WSLAST(R5)     ;AT THE END YET?
                   0D    1B    08CB  1561          BLEQU   110$                    ;CONTINUE
                51   0E A5    3C  08CD  1562          MOVZWL  PHD$W_WSDYN(R5),R1      ;BACK TO THE TOP
                04 0000'CF    E8  08D1  1563          BLBS    W^MMG$GB_FREWFLGS,110$  ;IF SWAPPER REQUESTED, DON'T MOVE LAST
                12 A5    50   B0  08D6  1564          MOVW    R0,PHD$W_WSLAST(R5)     ;SHRINK WSLAST BACK OVER 0 WSLE'S
                                  08DA  1565                                          ;THIS IS SAFE BECAUSE WS IS FULL
                 52   6541    D0  08DA  1566 110$:     MOVL    (R5)[R1],R2             ;R2 = VA FROM WSLE
                        E5    13  08DE  1567          BEQL    100$                    ;BRANCH IF UNUSABLE FREE ENTRY
                     8C 52    E9  08E0  1568 120$:     BLBC    R2,WSLENOVAL            ;BRANCH IF ENTRY NOT VALID
                8C 52    05    E0  08E3  1569          BBS     #WSL$V_WSLOCK,R2,50$    ;SKIP ENTRY IF IT IS LOCKED
                                  08E7  1570                                          ;CONDITION CODES STILL SET FROM LOAD OF R2
             08 0000'CF    00'  E1  08E7  1571          BBC     S^#EXE$V_TBCHK,W^EXE$GL_FLAGS,130$ ;BRANCH IF TBCHK NOT ENABLED
                   3F    52    DA  08ED  1572          MTPR    R2,#PR$_TBCHK           ;TEST FOR VALID IN TB
                         03    1C  08F0  1573          BVC     130$                    ;BRANCH IF NO VALID TRANSLATION
                      98 53    F4  08F2  1574          SOBGEQ  R3,60$                  ;SKIP PAGE UNLESS COUNT EXHAUSTED
                10 A5    51    B0  08F5  1575 130$:     MOVW    R1,PHD$W_WSNEXT(R5)     ;UPDATE NEXT POINTER
                      F704'   30  08F9  1576          BSBW    MMG$SVAPTECHK           ;RETURN R3 = SYS VA OF PAGE TABLE ENTRY
                                  08FC  1577                                          ;OK FOR PROCESS PAGE TABLES AND
                                  08FC  1578                                          ;PROCESS HEADER PAGES WITH PROCESS PCB ADR
                                  08FC  1579          .DISABLE LSB
                                  08FC  1580 ;
                                  08FC  1581 ; R1 = WSLX, R2 = VA FROM WSLE, R3 = SVAPTE, R4 = PCB, R5 = PHD
                                  08FC  1582 ;
                15 63    1A    E1  08FC  1583          BBC     #PTE$V_MODIFY,(R3),MMG$FREWSLX  ;BRANCH IF PAGE NOT MODIFIED
                                  0900  1584                                          ;IF ENTRY NOT VALID MODIFY=0 SO BRANCH
           0000'CF    0000'CF    D1  0900  1585          CMPL    W^MPW$GL_WAITLIM,W^SCH$GL_MFYCNT ;ABOVE WAIT PROCESS THRESHOLD?
                         0C    14  0907  1586          BGTR    MMG$FREWSLX             ;BRANCH IF SO
             06 0000'CF    00'  E0  0909  1587          BBS     S^#MMG$V_NOWAIT,W^MMG$GB_FREWFLGS,MMG$FREWSLX
                                  090F  1588                                          ;BRANCH IF THIS IS SWAPPER
                51    0C    3C  090F  1589          MOVZWL  #RSN$_MPWBUSY,R1        ;R1 = RESOURCE TO WAIT FOR
                      50    D4  0912  1590          CLRL    R0                      ;RETURN FAILURE INDICATION
                            05  0914  1591          RSB                             ;RETURN RESOURCE TO WAIT FOR IN R1
                                  0915  1592
                                  0915  1593          .ENABLE LSB
                                  0915  1594
                                  0915  1595 MMG$FREWSLX::
                         51    DD  0915  1596          PUSHL   R1                      ;SAVE WSLX FOR DELETE BY WSLX
                                  0917  1597
                                  0917  1598          ASSUME PTE$V_MODIFY EQ PTE$V_TYP1
     50    63    7B800000  8F    CB  0917  1599          BICL3   #^C<PTE$M_VALID ! -     ;FETCH VALID BIT
                                  091F  1600                  PTE$M_TYPT ! PTE$M_TYP0 ! - ;PTE TYPE BITS
                                  091F  1601                  PTE$M_GPTX>,(R3),R0     ;AND PFN/GPTX FROM PAGE TABLE ENTRY
                   0C 50    1F    E4  091F  1602          BBSC    #PTE$V_VALID,R0,10$     ;BRANCH IF PTE VALID
                                  0923  1603                                          ;CLEAR VALID BIT IN R0
                                  0923  1604 ;
                                  0923  1605 ; PAGE TABLE ENTRY NOT VALID,
                                  0923  1606 ;       PAGE IN TRANSITION AND READINPROG OR
                                  0923  1607 ;       GLOBAL PTE POINTING TO TRANSITION PTE
                                  0923  1608 ;
           51    50    EA  8F    78  0923  1609          ASHL    #-PTE$V_TYP0,R0,R1      ;IF NEITHER TYP1 OR TYP0 IS SET
                         18    13  0928  1610          BEQL    30$                     ;BRANCH TO RELEASE PAGE
                                  092A  1611 ;
                                  092A  1612 ; THIS WORKING SET LIST ENTRY POINTED TO A PAGE WITH A PAGE TABLE ENTRY
                                  092A  1613 ; WHICH IS NEITHER VALID NOR IN TRANSITION.  THIS PAGE MUST BE A GLOBAL
                                  092A  1614 ; PAGE ON THE WAY IN TO MEMORY.  THE GLOBAL PTE MUST BE IN TRANSITION.
                                  092A  1615 ;
                      FE97    30  092A  1616          BSBW    FRE_GBLTRANS            ;GET PFN IF PAGE IS GLOBAL TRANSITION
```

```
                      13      11  092D          BRB     30$                          ;RELEASE ACTIVE PAGE
                                  092F
             07 50    1A      E5  092F  10$:    BBCC    #PTE$V_MODIFY,R0,20$         ;CLR MODIFY BIT IN R0, BR IF CLR
     0000'DF40    80 8F       88  0933          BISB    #PFN$M_MODIFY,@W^PFN$AB_STATE[R0] ;RECORD MODIFY BIT
                                  093A  MPH$INVALIDHK::                              ;MULTI-PROCESSING CODE HOOKS IN HERE
          03 A3    84 8F      8A  093A  20$:    BICB    #<PTE$M_VALID ! PTE$M_MODIFY>@-24,3(R3) ;RESET VALID AND MODIFY
                                  093F                                               ;LEAVING TRANSITION PAGE FOR I/O TO SEE
                                  093F          INVALID R2                           ;INVALIDATE TRANSLATION BUFFER
               3A     52      DA  093F          MTPR    R2,S^#PR$_TBIS
                                  0942  MMG$FRE_TRYSKIP::                            ;HOOK RETURN LOCATION FOR MP SUPPORT
                                  0942  ;
                                  0942  ; RELEASE THIS WORKING SET LIST ENTRY
                                  0942  ;
        51   0000'DF40        D0  0942  30$:    MOVL    @W^PFN$AL_BAK[R0],R1         ;GET BACKING STORE ADDRESS (VBN)
             53 51    15      E0  0948          BBS     #PTE$V_CHKPNT,R1,90$         ;BRANCH IF CHECKPOINTABLE
     0A 0000'DF40     07      E0  094C  40$:    BBS     #PFN$V_MODIFY,@W^PFN$AB_STATE[R0],50$ ;BRANCH IF PAGE MODIFIED
          51     51   09      78  0953          ASHL    #32-PFN$S_BAK,R1,R1          ;DOES PAGE HAVE BACKING STORE?
                      0B      12  0957          BNEQ    70$                          ;BRANCH IF YES
                                  0959  ;
                                  0959  ; NULL PAGE FILE BACKING STORE ADDRESS, AND PAGE IS NOT MODIFIED
                                  0959  ;
                                  0959  NULLPGFL_NOMFY:
                                  0959          BUG_CHECK MFYNULPGFL,FATAL
                            FEFF  0959                  .WORD   ^XFEFF
                           0004'  095B                  .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_MFYNULPGFL!4
                                  095D  ;
                                  095D  ; NOW THAT WE HAVE A MODIFIED COPY OF THE PAGE IN MEMORY, NO NEED FOR OBSOLETE
                                  095D  ; PAGE FILE COPY.
                                  095D  ;
                      53      DD  095D  50$:    PUSHL   R3                           ;SAVE SVAPTE
                   F69E'      30  095F          BSBW    MMG$DALCBAKSTORE             ;RELEASE OLD PAGE FILE BACKING STORE
                      08      BA  0962          POPR    #^M<R3>                      ;RESTORE SVAPTE
                                  0964  ;
                                  0964  ; PFN$AL_BAK[R0] IS ALL SET UP, R0 = PFN, R2 = VA, R3 = SVAPTE (SLAVE IF GBL)
                                  0964  ;
                                  0964  70$:    ASSUME  PFN$C_PROCESS EQ 0
                                  0964          ASSUME  PFN$C_SYSTEM  EQ 1
                                  0964          ASSUME  PFN$C_GLOBAL  EQ 2
                                  0964          ASSUME  PFN$C_GBLWRT  EQ 3
                                  0964          ASSUME  PFN$C_PPGTBL  EQ 4
                                  0964          ASSUME  PFN$C_GPGTBL  EQ 5
                                  0964
   02     52     03   01      EC  0964          CMPV    #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,#PFN$C_GLOBAL ;GLOBAL PAGE?
                      6E      1F  0969          BLSSU   130$                         ;BRANCH IF PROCESS OR SYSTEM
                      5C      19  096B          BLSS    120$                         ;BRANCH IF PROCESS OR GLOBAL PAGE TABLE
                                  096D  ;
                                  096D  ; GLOBAL PAGE - MAKE SLAVE PTE INTO GLOBAL FORMAT
                                  096D  ;
 51   0000'DF40   0000'CF     C3  096D          SUBL3   W^MMG$GL_GPTBASE,@W^PFN$AL_PTE[R0],R1 ;BYTE INDEX TO GPTE
          51     51   1E      9C  0976          ROTL    #32-2,R1,R1                  ;GLOBAL PAGE TABLE INDEX
                                  097A          ASSUME  PTE$V_TYP0 EQ PTE$S_GPTX     ;TYP0 ADJACENT TO GPTX FIELD
             00 51    16      E2  097A          BBSS    #PTE$V_TYP0,R1,80$           ;SET TYP0 BIT FOR GLOBAL FORMAT
   63     17     00   51      F0  097E  80$:    INSV    R1,#PTE$V_GPTX,#PTE$S_GPTX+1,(R3) ;STORE GPTX + TYP0 IN PTE
                                  0983                                               ;CHANGING FROM TRANSITION TO GLOBAL
                    01B3      30  0983          BSBW    MMG$DECPTREF                 ;SLAVE PTE NO LONGER LOCKED
                                  0986          DECSHR  GTR=150$,-                   ;ONE LESS SHARER, BRANCH IF STILL IN USE
                                  0986                  IMAGE_FLAG=SYS_NONPAGED
```

```
                                                C 8
                                0986                    .SAVE_PSECT     LOCAL_BLOCK
                                0000000C                .PSECT  Z$INIT$PFN_FIXUP_TABLE
                     00000986'  000C                    .ADDRESS        ...PFN
                                B7  0010                .BYTE   OP$_DECW
                                D7  0011                .BYTE   OP$_DECL
                                00000986                .RESTORE_PSECT
                0000'DF40       B7  0986                DECW    @W^PFN$Ax_SHRCNT[R0]
                        6F      14  098B                    BGTR    150$
                        03      18  098D                    BGEQ    30006$
                    F66E'       30  098F                    BSBW    MMG$SHRCNTNEG
                                0992            30006$:
        51      0000'DF40       D0  0992  1671            MOVL    @W^PFN$AL_PTE[R0],R1    ;GET MASTER PTE ADR
         03 A1      84 8F       8A  0998  1672            BICB    #<PTE$M_VALID ! PTE$M_MODIFY>@-24,3(R1) ;FORM TRANSITION PTE
                     3A         11  099D  1673            BRB     130$                    ;GO COUNT ONE LESS WSL REF
                                099F  1674
                                099F  1675 ;  SEE IF OLD BACKING STORE ADDRESS SHOULD BE FORGOTTEN AND BAK REINITIALIZED
                                099F  1676
            0F 51       16      E0  099F  1677 90$:        BBS     #PTE$V_TYP0,R1,100$     ;BRANCH IF SECTION PAGE
            51      1F A5       9A  09A3  1678            MOVZBL  PHD$B_PAGFIL(R5),R1     ;MUST CHECK FOR LARGE PAGING FILE
        51      0000'DF41       D0  09A7  1679            MOVL    @W^MMG$GL_PAGSWPVC[R1],R1 ;GET ADDR OF PAGE FILE CONTROL BLOCK
          0E 1C A1              15  09AD  1680            BBS     #PTE$V_CHKPNT,PFL$L_MAXVBN(R1),110$;BRANCH IF PART OF LARGE VBN
    07 0000'DF40                07  09B2  1681 100$:      BBC     #PFN$V_MODIFY,@W^PFN$AB_STATE[R0],110$;BRANCH IF UNMODIFIED PAGE
    0000'DF40           1C A5   D0  09B9  1682            MOVL    PHD$L_PAGFIL(R5),@W^PFN$AL_BAK[R0] ;RESET BACKING STORE ADDRESS
        51      0000'DF40       D0  09C0  1683 110$:      MOVL    @W^PFN$AL_BAK[R0],R1    ;GET BACKING STORE ADDRESS
                    FF83        31  09C6  1684            BRW     40$                     ;CONTINUE
                                09C9  1685
                                09C9  1686 ;
                                09C9  1687 ; PROCESS OR GLOBAL PAGE TABLE
                                09C9  1688 ;
        04      52      03      01  ED  09C9  1689 120$:    CMPZV   #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,#PFN$C_PPGTBL ;PROCESS PAGE TABLE?
                                09      12  09CE  1690            BNEQ    130$                    ;BRANCH IF NO
                    51      42 A5       3C  09D0  1691            MOVZWL  PHD$W_PHVINDEX(R5),R1   ;PROCESS HEADER VECTOR INDEX
                     0000'DF41          B6  09D4  1692            INCW    @W^PHV$GL_REFCBAS[R1]   ;ADD A PROCESS HEADER REFERENCE
                                            09D9  1693                                    ;WHEN PROCESS PAGE TABLE IS PUT
                                            09D9  1694                                    ;INTO TRANSITION STATE
                                            09D9  1695 130$:    DECREF  EQL=140$                ;COUNT ONE LESS WSL REF
                0000'DF40           B7      09D9                        DECW    @W^PFN$AW_REFCNT[R0]
                        19          13      09DE                        BEQL    140$
                        03          18      09E0                        BGEQ    30009$
                    F61B'           30      09E2                        BSBW    MMG$REFCNTNEG
                                            09E5                30009$:
                                            09E5  1696 ;
                                            09E5  1697 ; OTHER REFERENCES OUTSTANDING, COULD BE DIRECT I/O, PAGING I/O
                                            09E5  1698 ; IF CURRENT PAGE STATE IS "ACTIVE" CHANGE IT TO "RELEASE PENDING" WHICH
                                            09E5  1699 ; IF REFAULTED WILL BE TRANSFORMED BACK TO ACTIVE.
                                            09E5  1700 ; LEAVE "READ IN PROGRESS" STATE AS IS.
                                            09E5  1701 ;
    07      0000'DF40       03      00  ED  09E5  1702            CMPZV   #PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0],#PFN$C_ACTIVE
                                            09ED  1703                                    ;UNLESS STATE IS "ACTIVE"
                                    0D  12  09ED  1704            BNEQ    150$                    ;LEAVE IT AS IT WAS
    0000'DF40           03      00  03  F0  09EF  1705            INSV    #PFN$C_RELPEND,#PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0]
                                            09F7  1706                                    ;OTHERWISE SET RELEASE PENDING STATE
                                    03  11  09F7  1707            BRB     150$
                                            09F9  1708 ;
                                            09F9  1709 ; R0 = PFN, REFCNT = 0, R2 = VA, R3 = SVAPTE
                                            09F9  1710 ;
                                            09F9  1711
```

```
                F604'  30  09F9  1712 140$:   BSBW    MMG$RELPFN              ;REFCNT = 0, RELEASE PFN
          51     8E   D0  09FC  1713 150$:   MOVL    (SP)+,R1                ;RECOVER SAVED WSLX
                 04   10  09FF  1714         BSBB    MMG$DELWSLEX            ;DELETE WORKING SET LIST ENTRY (BY INDEX)
          50     01   3C  0A01  1715         MOVZWL  #SS$_NORMAL,R0          ;SUCCESSFUL RETURN INDICATION
                      05  0A04  1716         RSB
                          0A05  1717
                          0A05  1718         .DISABLE LSB
```

```
0A05  1720          .SBTTL  DELWSLEX - DELETE WORKING SET LIST ENTRY BY INDEX
0A05  1721 ;++
0A05  1722 ; FUNCTIONAL DESCRIPTION:
0A05  1723 ;
0A05  1724 ;       THIS ROUTINE DELETES THE WORKING SET LIST ENTRY INDEXED BY
0A05  1725 ; R1, AND PLACES THE WORKING SET LIST ENTRY ON THE FREE LIST.
0A05  1726 ;
0A05  1727 ; CALLING SEQUENCE:
0A05  1728 ;
0A05  1729 ;       BSBW    MMG$DELWSLEX
0A05  1730 ;       BSBW    MMG$DELWSLEPPG
0A05  1731 ;
0A05  1732 ; INPUT PARAMETERS:
0A05  1733 ;
0A05  1734 ;       R1 = WORKING SET LIST INDEX
0A05  1735 ;       R2 = VIRTUAL ADDRESS IF ENTERING AT DELWSLEPPG
0A05  1736 ;       R4 = PROCESS CONTROL BLOCK ADDRESS
0A05  1737 ;       R5 = PROCESS HEADER ADDRESS
0A05  1738 ;
0A05  1739 ; IMPLICIT INPUTS:
0A05  1740 ;
0A05  1741 ;       NONE
0A05  1742 ;
0A05  1743 ; OUTPUT PARAMETERS:
0A05  1744 ;
0A05  1745 ;       R0 PRESERVED
0A05  1746 ;
0A05  1747 ; IMPLICIT OUTPUTS:
0A05  1748 ;
0A05  1749 ;       NONE
0A05  1750 ;
0A05  1751 ; COMPLETION CODES:
0A05  1752 ;
0A05  1753 ;       NONE
0A05  1754 ;
0A05  1755 ; SIDE EFFECTS:
0A05  1756 ;
0A05  1757 ;       NONE
0A05  1758 ;
0A05  1759 ;--
```

```
                                0A05  1761          .ENABL  LSB
                                0A05  1762
                                0A05  1763 MMG$DELWSLEX::
                          52    0A05  1764          MOVL    (R5)[R1],R2                 ;FETCH WORKING SET LIST ENTRY
          53    52    03      0A09  1765          EXTZV   #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,R3  ;GET THE PAGE TYPE
                                0A0E  1766          CASE    R3,<-                       ;AND DISPATCH ON IT
                                0A0E  1767                  30$,-                       ;PROCESS PAGE
                                0A0E  1768                  45$,-                       ;SYSTEM PAGE
                                0A0E  1769                  10$,-                       ;GLOBAL READ ONLY
                                0A0E  1770                  10$,-                       ;GLOBAL WRITABLE
                                0A0E  1771                  45$,-                       ;PROCESS PAGE TABLE
                                0A0E  1772                  45$  >                      ;GLOBAL PAGE TABLE
          05'   00    53    AF  0A0E          CASEW   R3,#0,S^#<<30011$-30010$>/2>-1
                                0A12      30010$:
                          0017' 0A12          .SIGNED_WORD    30$-30010$
                          0019' 0A14          .SIGNED_WORD    45$-30010$
                          0010' 0A16          .SIGNED_WORD    10$-30010$
                          0010' 0A18          .SIGNED_WORD    10$-30010$
                          0019' 0A1A          .SIGNED_WORD    45$-30010$
                          0019' 0A1C          .SIGNED_WORD    45$-30010$
                                0A1E      30011$:
                                0A1E  1773          BUG_CHECK DELWSLEX,FATAL            ;BAD PAGE TYPE
                          FEFF  0A1E                  .WORD   ^XFEFF
                          0004' 0A20                  .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_DELWSLEX!4
                                0A22  1774 ;
                                0A22  1775 ; GLOBAL PAGE, READ ONLY OR WRITABLE
                                0A22  1776 ;
                          0E    0A22  1777 10$:     BSBB    DECVALWSLECNT               ;DEC VALID WORKING SET LIST ENTRY COUNT
                    34    A4    0A24  1778          DECW    PCB$W_GPGCNT(R4)            ;ONE LESS GLOBAL PAGE IN WORKING SET
                          05    0A27  1779          BRB     50$
                                0A29  1780 ;
                                0A29  1781 ; PROCESS PAGE, R2 = VIRTUAL ADDRESS
                                0A29  1782 ;
                                0A29  1783 MMG$DELWSLEPPG::
                          07    0A29  1784 30$:     BSBB    DECVALWSLECNT               ;DECREMENT VALID WORKING SET LIST ENTRY CNT
                    36    A4    0A2B  1785 45$:     DECW    PCB$W_PPGCNT(R4)            ;ONE LESS PROCESS PAGE IN WORKING SET
                     6541       0A2E  1786 50$:     CLRL    (R5)[R1]                    ;FREE THE WORKING SET LIST ENTRY
                          05    0A31  1787          RSB
                                0A32  1788
                                0A32  1789          .DSABL  LSB
                                0A32  1790
                                0A32  1791 DECVALWSLECNT:
          53    55    68    A5  0A32  1792          ADDL3   PHD$L_PTWSLEVAL(R5),R5,R3 ;BASE ADR OF BYTE ARRAY OF COUNTS OF
                                0A37  1793                                              ;VALID WSLE'S IN EACH PAGE TABLE
    52    52    0F    10    EE  0A37  1794          EXTV    #VA$V_VPN+7,#VA$S_VPN+1-7,R2,R2 ;BITS 16:30 OF VA SIGN EXTENDED
                      05    18  0A3C  1795          BGEQ    10$                         ;BRANCH IF P0 SPACE
          53    0000'CF       C0  0A3E  1796          ADDL    W^SGN$GL_PTPAGCNT,R3        ;END ADDRESS OF BYTE ARRAY
                       6342     97  0A43  1797 10$:     DECB    (R3)[R2]                ;ONE LESS VALID WSLE IN THIS PAGE TABLE
                          03    18  0A46  1798          BGEQ    20$                         ;BRANCH IF PT STILL HAS OTHER VALID WSLE'S
                    6E    A5    B7  0A48  1799          DECW    PHD$W_PTCNTVAL(R5)          ;ONE LESS PT WITH VALID WSLE'S
                          05    0A4B  1800 20$:     RSB
```

```
                        .SBTTL  ININEWPFN - ALLOCATE AND INIT A NEW PFN
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       ALLOCATE A NEW PFN AND INITIALIZE THE PFN DATA BASE FOR IT
                ; AND MAKE A WORKING SET LIST ENTRY.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW    MMG$ININEWPFN           ;ALLOCATE AND INIT NEW PFN
                ;
                ; INPUT PARAMETERS:
                ;
                ;       R2 = FAULT VA (LOW BITS = PAGTYP)
                ;       R3 = SVAPTE (SLAVE IF GLOBAL)
                ;       R4 = PCB ADDRESS (PROCESS IF GLOBAL)
                ;       R5 = PROCESS HEADER ADDRESS (PROCESS IF GLOBAL)
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       R0 = PFN, OR NEGATIVE IF NONE AVAILABLE
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       NONE
                ;
                ; COMPLETION CODES:
                ;
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;
                ;       NONE
                ;
                ;--
```

```
                                0A4C  1842 ININEWPFNWAIT:
                     2C   BA    0A4C  1843          POPR    #^M<R2,R3,R5>                   ;RESTORE SAVED REGISTERS
                     05   0A4E  1844          RSB                                     ;AND RETURN NO FREE PAGES INDICATION
                                0A4F  1845
                                0A4F  1846 MMG$ININEWPFN::
                     55   DD    0A4F  1847          PUSHL   R5                      ;SAVE PHD
                     53   DD    0A51  1848          PUSHL   R3                      ;SAVE SVAPTE
                     52   DD    0A53  1849          PUSHL   R2                      ;SAVE VA
                 FSA8'    30    0A55  1850          BSBW    MMG$ALLOCPFN            ;ALLOCATE A NEW PFN
             F0 50    1F   E0    0A58  1851          BBS     #31,R0,ININEWPFNWAIT    ;BRANCH IF NONE AVAILABLE
                 52   6E   7D    0A5C  1852          MOVQ    (SP),R2                 ;R2=VA, R3=SVAPTE
          02   52   03   01    EC    0A5F  1853          CMPV    #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,#PFN$C_GLOBAL ;GLOBAL PAGE?
                    10   19    0A64  1854          BLSS    40$                     ;BRANCH IF NOT
          53   63   16   00    EF    0A66  1855          EXTZV   #PTE$V_GPTX,#PTE$S_GPTX,(R3),R3 ;GLOBAL PAGE TABLE INDEX
              53   0000'DF43    DE    0A6B  1856          MOVAL   @W^MMG$GL_GPTBASE[R3],R3 ;SVAPTE OF MASTER
               55   0000'DF    DE    0A71  1857          MOVAL   @W^MMG$GL_SYSPHD,R5     ;SYSTEM PROCESS HEADER ADR FOR GLOBAL
                 03 52   1F   E0    0A76  1858 40$:      BBS     #VA$V_SYSTEM,R2,50$     ;DON'T COUNT PT REF FOR SYSTEM PAGE
                       007E   30    0A7A  1859          BSBW    MMG$INCPTREF            ;LOCK PAGE TABLE ENTRY (NOT FOR SYSTEM)
                0000'DF40   53    D0    0A7D  1860 50$:      MOVL    R3,@W^PFN$AL_PTE[R0]    ;STORE PTE ADDRESS
                                0A83  1861          ASSUME  PFN$V_PAGTYP EQ 0
                52   52   1F   9C    0A83  1862          ROTL    #<32-WSL$V_PAGTYP>,R2,R2 ;POSITION PAGE TYPE FIELD
       0000'DF40    52    F8 8F    88    0A87  1863          BICB3   #^C<PFN$M_PAGTYP>,R2,@W^PFN$AB_TYPE[R0] ;SET PAGE TYPE
                    2C   BA    0A8F  1864          POPR    #^M<R2,R3,R5>                   ;R2=VA, R3=SVAPTE, R5=PHD
                                0A91  1865 ;
                                0A91  1866 ; FALL THROUGH TO MMG$MAKEWSLE
                                0A91  1867 ;
```

```
                                1 8
0A91  1869          .SBTTL  MAKEWSLE - MAKE A WORKING SET LIST ENTRY
0A91  1870 ;++
0A91  1871 ; FUNCTIONAL DESCRIPTION:
0A91  1872 ;
0A91  1873 ;       THIS ROUTINE ENTERS SPECIFIED VIRTUAL ADDRESS INTO THE WORKING
0A91  1874 ; SET LIST.  IT ASSUMES THAT THERE IS A FREE WORKING SET LIST ENTRY ON THE
0A91  1875 ; FREE LIST.  IF THE PAGE IS A GLOBAL PAGE THE SLAVE PAGE TABLE ENTRY IS
0A91  1876 ; LOCKED AT THIS TIME AND THE SHRCNT AND/OR REFCNT IS INCREMENTED.
0A91  1877 ; THIS ROUTINE ALSO KEEPS THE ACTIVE PAGE COUNTERS IN THE PCB (PPGCNT, GPGCNT).
0A91  1878 ;
0A91  1879 ;
0A91  1880 ; CALLING SEQUENCE:
0A91  1881 ;
0A91  1882 ;       BSBW    MMG$MAKEWSLE
0A91  1883 ;
0A91  1884 ; INPUT PARAMETERS:
0A91  1885 ;
0A91  1886 ;       R0 = PAGE FRAME NUMBER
0A91  1887 ;       R2 = VA (LOW BITS = PAGTYP)
0A91  1888 ;       R3 = SVAPTE (SLAVE IF GLOBAL)
0A91  1889 ;       R4 = PCB ADDRESS (PROCESS IF GLOBAL)
0A91  1890 ;       R5 = PHD ADDRESS (PROCESS IF GLOBAL)
0A91  1891 ;
0A91  1892 ; IMPLICIT INPUTS:
0A91  1893 ;
0A91  1894 ;       FREE WORKING SET LIST CONTAINS AT LEAST ONE ENTRY
0A91  1895 ;
0A91  1896 ; OUTPUT PARAMETERS:
0A91  1897 ;
0A91  1898 ;       R0 = PFN PRESERVED
0A91  1899 ;
0A91  1900 ; IMPLICIT OUTPUTS:
0A91  1901 ;
0A91  1902 ;       NONE
0A91  1903 ;
0A91  1904 ; COMPLETION CODES:
0A91  1905 ;
0A91  1906 ;       NONE
0A91  1907 ;
0A91  1908 ; SIDE EFFECTS:
0A91  1909 ;
0A91  1910 ;       NONE
0A91  1911 ;
0A91  1912 ;--
```

```
                          0A91  1914 MMG$MAKEWSLE::
                    51  10 A5  3C  0A91  1915          MOVZWL  PHD$W_WSNEXT(R5),R1     ;WSLX FOR FREE ENTRY
                          0A95  1916
                          0A95  1917          ASSUME  WSL$V_VALID EQ 0
                       1A 6541  E8  0A95  1918          BLBS    (R5)[R1],20$            ;BRANCH IF ENTRY BUSY, ERROR
                  6541  52  01  C9  0A99  1919          BISL3   #WSL$M_VALID,R2,(R5)[R1] ;STORE NEW WSLE
                          0A9E  1920
               51  52  03  01  EF  0A9E  1921          EXTZV   #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,R1 ;EXTRACT THE PAGE TYPE
                          0AA3  1922          CASE    R1,<-                           ;AND DISPATCH ON IT
                          0AA3  1923                  50$,-                           ;PROCESS PAGE
                          0AA3  1924                  70$,-                           ;SYSTEM PAGE
                          0AA3  1925                  30$,-                           ;GLOBAL READ ONLY
                          0AA3  1926                  30$,-                           ;GLOBAL WRITABLE
                          0AA3  1927                  70$,-                           ;PROCESS PAGE TABLE
                          0AA3  1928                  70$  >                          ;GLOBAL PAGE TABLE
                05'  00  51  AF  0AA3          CASEW   R1,#0,S^#<<30013$-30012$>/2>-1
                          0AA7          30012$:
                     0022'  0AA7          .SIGNED_WORD    50$-30012$
                     0024'  0AA9          .SIGNED_WORD    70$-30012$
                     0010'  0AAB          .SIGNED_WORD    30$-30012$
                     0010'  0AAD          .SIGNED_WORD    30$-30012$
                     0024'  0AAF          .SIGNED_WORD    70$-30012$
                     0024'  0AB1          .SIGNED_WORD    70$-30012$
                          0AB3          30013$:
                          0AB3  1929 20$:       BUG_CHECK MAKEWSLE,FATAL        ;BAD PAGE TYPE OR
                     FEFF  0AB3                 .WORD   ^XFEFF
                     0004' 0AB5                 .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_MAKEWSLE!4
                          0AB7  1930                                            ;WSNEXT POINTS TO VALID WSLE
                          0AB7  1931 ;
                          0AB7  1932 ; GLOBAL PAGE, READ ONLY OR WRITABLE
                          0AB7  1933 ;
                    42  10  0AB7  1934 30$:       BSBB    MMG$INCPTREF            ;LOCK THE SLAVE PAGE TABLE ENTRY
                    20  10  0AB9  1935          BSBB    INCVALWSLECNT           ;INC VALID WORKING SET LIST ENTRY COUNT
                 34 A4  B6  0ABB  1936          INCW    PCB$W_GPGCNT(R4)        ;ANOTHER GLOBAL PAGE IN WORKING SET LIST
                          0ABE  1937                  PFN_REFERENCE   -
                          0ABE  1938          ACBW    <#1,#1,@W^PFN$Ax_SHRCNT[R0],80$>,-      ;COUNT SHARER, BRANCH IF FIR
                          0ABE  1939                  LONG_OPCODE=ACBL,-
                          0ABE  1940                  IMAGE=SYS_NONPAGED
                          0ABE                .SAVE_PSECT     LOCAL_BLOCK
                     00000012                 .PSECT  Z$INIT$PFN_FIXUP_TABLE
                00000ABE' 0012                .ADDRESS        ...PFN
                       3D 0016                .BYTE   OP$_ACBW
                       F1 0017                .BYTE   OP$_ACBL
                     00000ABE                 .RESTORE_PSECT
000E 0000'DF40  01  01  3D  0ABE          ACBW    #1,#1,@W^PFN$Ax_SHRCNT[R0],80$
                    11  11  0AC7  1941          BRB     90$
                          0AC9  1942 ;
                          0AC9  1943 ; PROCESS PAGE
                          0AC9  1944 ;
                    10  10  0AC9  1945 50$:       BSBB    INCVALWSLECNT           ;INC VALID WORKING SET LIST ENTRY COUNT
                 36 A4  B6  0ACB  1946 70$:       INCW    PCB$W_PPGCNT(R4)        ;ONE MORE ACTIVE PROCESS PAGE IN WSL
                          0ACE  1947                  PFN_REFERENCE   -
                          0ACE  1948          MOVW    <PHD$W_WSNEXT(R5),@W^PFN$Ax_WSLX[R0]>,- ;SET INDEX TO WSLE
                          0ACE  1949                  LONG_OPCODE=MOVZWL,-
                          0ACE  1950                  IMAGE=SYS_NONPAGED
                          0ACE                .SAVE_PSECT     LOCAL_BLOCK
                     00000018                 .PSECT  Z$INIT$PFN_FIXUP_TABLE
```

```
              00000ACE' 0018            .ADDRESS        ...PFN
                    B0  001C            .BYTE   OP$_MOVW
                    3C  001D            .BYTE   OP$_MOVZWL
                    00000ACE            .RESTORE_PSECT
0000'DF40    10 A5  B0  0ACE            MOVW    PHD$W_WSNEXT(R5),@W^PFN$Ax_WSLX[R0]
                        0AD5  1951                                              ;ONLY FOR PRIVATE PAGES
                        0AD5  1952
        0000'DF40   B6  0AD5  1953 80$:  INCW    @W^PFN$AW_REFCNT[R0]      ;ANOTHER REFERENCE FOR THE PAGE
                    05  0ADA  1954 90$:  RSB
                        0ADB  1955
                        0ADB  1956 INCVALWSLECNT:
  53    55   68 A5  C1  0ADB  1957          ADDL3   PHD$L_PTWSLEVAL(R5),R5,R3 ;BASE ADR OF BYTE ARRAY OF COUNTS OF
                        0AE0  1958                                          ;VALID WSLE'S IN EACH PAGE TABLE
52   52   0F    10  EE  0AE0  1959          EXTV    #VA$V_VPN+7,#VA$S_VPN+1-7,R2,R2 ;BITS 16:30 OF VA SIGN EXTENDED
                05  18  0AE5  1960          BGEQ    10$                       ;BRANCH IF P0 SPACE
     53   0000'CF   C0  0AE7  1961          ADDL    W^SGN$GL_PTPAGCNT,R3      ;BASE ADR TO NEGATIVE INDEX FROM
              6342  96  0AEC  1962 10$:     INCB    (R3)[R2]                  ;ANOTHER VALID WSLE IN THIS PAGE TABLE
                03  14  0AEF  1963          BGTR    20$                       ;BRANCH IF NOT THE FIRST
             6E A5  B6  0AF1  1964          INCW    PHD$W_PTCNTVAL(R5)        ;ANOTHER PAGE TABLE WITH VALID WSLE'S
                    05  0AF4  1965 20$:     RSB
```

```
                        0AF5  1967          .SBTTL  LOCKPGTB - LOCK PAGE TABLE
                        0AF5  1968 ;++
                        0AF5  1969 ; FUNCTIONAL DESCRIPTION:
                        0AF5  1970 ;
                        0AF5  1971 ;       LOCKPGTB TAKES A VIRTUAL ADDRESS, REFERENCES AND
                        0AF5  1972 ;       LOCKS THE ASSOCIATED PAGE TABLE, AND RETURNS THE SYSTEM
                        0AF5  1973 ;       VIRTUAL ADDRESS OF THE PAGE TABLE ENTRY.  IT IS CALLED
                        0AF5  1974 ;       WITH IPL = ASTDEL OR LOWER AND RETURNS WITH IPL = SYNCH.
                        0AF5  1975 ;
                        0AF5  1976 ; CALLING SEQUENCE:
                        0AF5  1977 ;
                        0AF5  1978 ;       BSBW    MMG$LOCKPGTB
                        0AF5  1979 ;
                        0AF5  1980 ; INPUT PARAMETERS:
                        0AF5  1981 ;
                        0AF5  1982 ;       R2 = VIRTUAL ADDRESS
                        0AF5  1983 ;       R4 = PROCESS CONTROL BLOCK ADDRESS
                        0AF5  1984 ;       R5 = PROCESS HEADER ADDRESS (P1 SPACE IF PROCESS PCB,
                        0AF5  1985 ;            SYSTEM SPACE OF SYSTEM PCB)
                        0AF5  1986 ;       IPL = ASTDEL OR LOWER
                        0AF5  1987 ;
                        0AF5  1988 ; IMPLICIT INPUTS:
                        0AF5  1989 ;
                        0AF5  1990 ;       NONE
                        0AF5  1991 ;
                        0AF5  1992 ; OUTPUT PARAMETERS:
                        0AF5  1993 ;
                        0AF5  1994 ;       R2 PRESERVED
                        0AF5  1995 ;       R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                        0AF5  1996 ;       IPL = SYNCH
                        0AF5  1997 ;
                        0AF5  1998 ; IMPLICIT OUTPUTS:
                        0AF5  1999 ;
                        0AF5  2000 ;       PAGE TABLE LOCKED VIA INCPTREF
                        0AF5  2001 ;
                        0AF5  2002 ; COMPLETION CODES:
                        0AF5  2003 ;
                        0AF5  2004 ;       NONE
                        0AF5  2005 ;
                        0AF5  2006 ; SIDE EFFECTS:
                        0AF5  2007 ;
                        0AF5  2008 ;       NONE
                        0AF5  2009 ;
                        0AF5  2010 ;--
                        0AF5  2011
                        0AF5  2012 MMG$LOCKPGTB::
 F508'  30              0AF5  2013          BSBW    MMG$PTEREF              ;REFERENCE PTE, GET SVAPTE
                        0AF8  2014                                          ;RETURNS AT IPL=SYNCH
3A 50   E9              0AF8  2015          BLBC    R0,INCPTREFBUG          ;BRANCH IF LENGTH VIOLATION
                        0AFB  2016 ;
                        0AFB  2017 ; FALL THROUGH TO MMG$INCPTREF
                        0AFB  2018 ;
```

```
                        .SBTTL  INCPTREF - INCREMENT PAGE TABLE REFERENCE COUNT
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE ACCEPTS THE ADDRESS OF A PAGE TABLE ENTRY AND
                ; LOCKS THE ASSOCIATED PAGE TABLE INTO MEMORY.  IT ALSO MAINTAINS THE
                ; COUNT OF SUCH LOCKED PAGE TABLES IN THE PROCESS HEADER VECTOR.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW    MMG$INCPTREF
                ;
                ; INPUT PARAMETERS:
                ;
                ;       R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY (MASTER IF GLOBAL)
                ;       R5 = PROCESS HEADER ADDRESS (SYSTEM IF GLOBAL)
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       R0,R2,R3 PRESERVED
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       NONE
                ;
                ; COMPLETION CODES:
                ;
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;
                ;       NONE
                ;
                ;--
```

```
                                0AFB  2059 MMG$INCPTREF::
          51   53   15   09   EF  0AFB  2060          EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;GET PAGE NUMBER OF PT CONTAINING THIS PTE
                                0B00  2061 ;
                                0B00  2062 ; ***** WARNING ***** THE FOLLOWING DEPENDS ON GPTBASE = SPTBASE
                                0B00  2063 ;
             51   0000'DF41    D0  0B00  2064          MOVL    @W^MMG$GL_SPTBASE[R1],R1        ;PTE FOR PAGE TABLE
                         2D    18  0B06  2065          BGEQ    20$                             ;DISASTER IF NOT VALID
          51   51   15   00   EF  0B08  2066          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R1,R1 ;PAGE FRAME NUMBER
               0000'CF   51   D1  0B0D  2067          CMPL    R1,W^MMG$GL_MAXPFN              ; IS THERE PFN DATABASE?
                         09   14  0B12  2068          BGTR    5$                              ; NO, SKIP INCREMENT
                                0B14  2069          PFN_REFERENCE   -
                                0B14  2070          ACBW    <#1,#1,@W^PFN$Ax_SHRCNT[R1],10$>,-      ;INC SHRCNT, BRANCH IF FIRST
                                0B14  2071                  LONG_OPCODE=ACBL,-
                                0B14  2072                  IMAGE=SYS_NONPAGED
                                0B14                .SAVE_PSECT     LOCAL_BLOCK
                            0000001E                .PSECT  Z$INIT$PFN_FIXUP_TABLE
                    00000B14'  001E                 .ADDRESS                ...PFN
                          3D   0022                 .BYTE   OP$_ACBW
                          F1   0023                 .BYTE   OP$_ACBL
                            00000B14                .RESTORE_PSECT
0001 0000'DF41    01   01   3D  0B14                ACBW    #1,#1,@W^PFN$Ax_SHRCNT[R1],10$
                          05  0B1D  2073 5$:       RSB
                                0B1E  2074 ;
                                0B1E  2075 ; SHARE COUNT JUST WENT FROM 0 TO 1 INDICATING THAT THE FIRST ACTIVE
                                0B1E  2076 ; PAGE TABLE ENTRY WAS JUST PLACED IN THE PAGE TABLE
                                0B1E  2077 ;
                                0B1E  2078 ; ASSUMPTION HERE IS THAT THIS ROUTINE IS NOT CALLED FOR SYSTEM PAGE TABLES
                                0B1E  2079 ; THIS IS EITHER A PROCESS OR GLOBAL PAGE TABLE.
                                0B1E  2080 ;
                                0B1E  2081 10$:             PFN_REFERENCE   -
                                0B1E  2082          MOVZWL  <@W^PFN$Ax_WSLX[R1],R1>,-       ;WORKING SET LIST INDEX
                                0B1E  2083                  LONG_OPCODE=MOVL,-
                                0B1E  2084                  IMAGE=SYS_NONPAGED
                                0B1E                .SAVE_PSECT     LOCAL_BLOCK
                            00000024                .PSECT  Z$INIT$PFN_FIXUP_TABLE
                    00000B1E'  0024                 .ADDRESS                ...PFN
                          3C   0028                 .BYTE   OP$_MOVZWL
                          D0   0029                 .BYTE   OP$_MOVL
                            00000B1E                .RESTORE_PSECT
          51   0000'DF41    3C  0B1E                MOVZWL  @W^PFN$Ax_WSLX[R1],R1
               6541   20    C8  0B24  2085          BISL    #WSL$M_WSLOCK,(R5)[R1]  ;SET WORKING SET LOCKDOWN BIT
                   70 A5    B6  0B28  2086          INCW    PHD$W_PTCNTACT(R5)      ;ANOTHER ACTIVE PAGE TABLE
                51   42 A5   3C  0B2B  2087          MOVZWL  PHD$W_PHVINDEX(R5),R1   ;PROCESS HEADER VECTOR INDEX
                0000'DF41    B6  0B2F  2088          INCW    @W^PHV$GL_REFCBAS[R1]   ;COUNT ANOTHER PAGE TABLE LOCKED
                          05  0B34  2089          RSB
                                0B35  2090 ;
                                0B35  2091 ; PAGE TABLE PAGE WAS NOT VALID
                                0B35  2092 ;
                                0B35  2093 20$:
                                0B35  2094 INCPTREFBUG:
                                0B35  2095          BUG_CHECK INCPTREF,FATAL        ;PAGE TABLE NOT VALID
                            FEFF  0B35                      .WORD   ^XFEFF
                            0004' 0B37                      .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_INCPTREF!4
                                0B39  2096                                          ;LENGTH VIOLATION FROM LOCKPGTB
```

```
0B39          .SBTTL  DECPTREF - DECREMENT PAGE TABLE REFERENCE COUNT
0B39    ;++
0B39    ; FUNCTIONAL DESCRIPTION:
0B39    ;
0B39    ;       THIS ROUTINE DECREMENTS THE REFERENCE COUNT FOR THE PAGE TABLE
0B39    ; CONTAINING THE PAGE TABLE ENTRY ADDRESSED BY R3.  IF THE RESULTING REFERENCE
0B39    ; COUNT INDICATES THAT NO MORE PAGE TABLE ENTRIES ARE IN USE, THE PROCESS
0B39    ; HEADER VECTOR REFERENCE COUNT IS DECREMENTED AS WELL INDICATING A
0B39    ; FREE PAGE TABLE
0B39    ;
0B39    ; CALLING SEQUENCE:
0B39    ;
0B39    ;       BSBW    MMG$DECPTREF
0B39    ;
0B39    ; INPUT PARAMETERS:
0B39    ;
0B39    ;       R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
0B39    ;
0B39    ; IMPLICIT INPUTS:
0B39    ;
0B39    ;       NONE
0B39    ;
0B39    ; OUTPUT PARAMETERS:
0B39    ;
0B39    ;       R0,R2,R3 PRESERVED
0B39    ;
0B39    ; IMPLICIT OUTPUTS:
0B39    ;
0B39    ;       NONE
0B39    ;
0B39    ; COMPLETION CODES:
0B39    ;
0B39    ;       NONE
0B39    ;
0B39    ; SIDE EFFECTS:
0B39    ;
0B39    ;       NONE
0B39    ;
0B39    ;--
```

```
                                 0B39  MMG$DECPTREF::
      51   53   15   09    EF    0B39          EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;INDEX TO SPT ENTRY FOR PAGE TABLE
         51   0000'DF41    D0    0B3E          MOVL    @W^MMG$GL_SPTBASE[R1],R1          ;PTE FOR PAGE TABLE
                     56    18    0B44          BGEQ    40$                      ;BRANCH IF NOT VALID, ERROR
      51   51   15   00    EF    0B46          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R1,R1 ;PAGE FRAME NUMBER FOR PAGE TABLE
            0000'CF  51    D1    0B4B          CMPL    R1,W^MMG$GL_MAXPFN       ;IS THERE PFN DATA BASE FOR THIS PAGE?
                     49    14    0B50          BGTR    20$                      ;NO, SKIP DECREMENT
                                 0B52          DECSHR  PFN=R1,GTR=20$,-         ;ONE LESS ACTIVE PTE IN THIS PT
                                 0B52                  IMAGE_FLAG=SYS_NONPAGED
                                 0B52          .SAVE_PSECT      LOCAL_BLOCK
                           0000002A            .PSECT  Z$INIT$PFN_FIXUP_TABLE
                      00000B52' 002A           .ADDRESS         ...PFN
                           B7    002E          .BYTE   OP$_DECW
                           D7    002F          .BYTE   OP$_DECL
                           00000B52            .RESTORE_PSECT
            0000'DF41    B7    0B52            DECW    @W^PFN$Ax_SHRCNT[R1]
                     42    14    0B57                  BGTR    20$
                     03    18    0B59                  BGEQ    30022$
                  F4A2'    30    0B5B                  BSBW    MMG$SHRCNTNEG
                                 0B5E  30022$:
                                 0B5E  ;
                                 0B5E  ; SHARE COUNT JUST WENT TO 0, THIS PAGE TABLE IS NO LONGER REQUIRED
                                 0B5E  ; TO REMAIN RESIDENT
                                 0B5E  ; R1 = PFN FOR PAGE TABLE PAGE
                                 0B5E  ;
                     50    DD    0B5E          PUSHL   R0                       ;SAVE THIS REGISTER
                                 0B60                  PFN_REFERENCE   -
                                 0B60          MOVZWL  <@W^PFN$Ax_WSLX[R1],R0>,-       ;USE IT TO HOLD THE WORKING SET LIST
                                 0B60                  LONG_OPCODE=MOVL,-
                                 0B60                  IMAGE=SYS_NONPAGED
                                 0B60          .SAVE_PSECT      LOCAL_BLOCK
                           00000030            .PSECT  Z$INIT$PFN_FIXUP_TABLE
                      00000B60' 0030           .ADDRESS         ...PFN
                           3C    0034          .BYTE   OP$_MOVZWL
                           D0    0035          .BYTE   OP$_MOVL
                           00000B60            .RESTORE_PSECT
      50    0000'DF41    3C    0B60            MOVZWL  @W^PFN$Ax_WSLX[R1],R0
         51   0000'DF    DE    0B66            MOVAL   @W^MMG$GL_SYSPHD,R1      ;ADDRESS OF SYSTEM HEADER
              51   53    D1    0B6B            CMPL    R3,R1                    ;PTE ADR IN SYSTEM HEADER?
                     20    1E    0B6E          BGEQU   10$                      ;BRANCH IF YES, GLOBAL PAGE TABLE
      51   53   0000'CF    C3    0B70          SUBL3   W^SWP$GL_BALBASE,R3,R1   ;ADR RELATIVE TO BEGIN OF BAL SET
           51   0000'CF    C6    0B76          DIVL    W^SWP$GL_BSLOTSZ,R1      ;PROCESS HEADER INDEX
    51   51   F7   8F    78    0B7B            ASHL    #-9,R1,R1                ;DIVIDE BY PAGE SIZE
                     22    10    0B80          BSBB    MMG$DECPHDREF1           ;DECREMENT PROCESS HEADER REFERENCE COUNT
           51   0000'CF    C4    0B82          MULL    W^SWP$GL_BSLOTSZ,R1      ;CONVERT PROCESS HEADER INDEX
           51   51   09    9C    0B87          ROTL    #9,R1,R1                 ;MULL BY PAGE SIZE
           51   0000'CF    C0    0B8B          ADDL    W^SWP$GL_BALBASE,R1      ;TO PROCESS HEADER ADDRESS
              6140   20    CA    0B90  10$:    BICL    #WSL$M_WSLOCK,(R1)[R0]   ;SHUT OFF WORKING SET LOCK
                  70 A1    B7    0B94          DECW    PHD$W_PTCNTACT(R1)       ;ONE LESS ACTIVE PAGE TABLE
                     03    19    0B97          BLSS    40$                      ;BRANCH IF ERROR
                     01    BA    0B99          POPR    #^M<R0>                  ;RESTORE SAVED REGISTER
                           05    0B9B  20$:    RSB                              ;AND RETURN
                                 0B9C  ;
                                 0B9C  ; PAGE TABLE PTE NOT VALID, OR PAGE TABLE REFERENCE COUNT IS BAD
                                 0B9C  ; OR PTCNTACT WENT NEGATIVE
                                 0B9C  ;
                                 0B9C  40$:    BUG_CHECK DECPTREF,FATAL         ;ERROR IN DECPTREF
```

```
                        FEFF  0B9C                      .WORD   ^XFEFF
                        0004' 0B9E                      .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_DECPTREF!4
```

```
0BA0  2178          .SBTTL  DECPHDREF - DECREMENT PROCESS HEADER REFERENCE COUNT
0BA0  2179 ;++
0BA0  2180 ; FUNCTIONAL DESCRIPTION:
0BA0  2181 ;
0BA0  2182 ;       DECPHDREF REDUCES THE PROCESS HEADER REFERENCE COUNT AND INFORMS
0BA0  2183 ;       THE SWAPPER IF THE COUNT GOES TO ZERO.  THIS COUNT IS RAISED ONCE
0BA0  2184 ;       FOR EACH REASON THAT A GIVEN SPT ENTRY IS BUSY NOT COUNTING THE
0BA0  2185 ;       WORKING SET LIST ENTRY REFERENCE.  THE FOLLOWING ARE REASONS WHY
0BA0  2186 ;       THE REFERENCE COUNT IS INCREASED FOR A GIVEN PAGE TABLE PAGE.
0BA0  2187 ;               1. PLACED ON THE FREE OR MODIFIED LIST
0BA0  2188 ;               2. READ OR WRITE IN PROGRESS
0BA0  2189 ;               3. SHARE COUNT IS ABOVE 0, I.E. IT CONTAINS ACTIVE PTE'S
0BA0  2190 ;       THE REFERENCE COUNT IS DECREASED UNDER THE FOLLOWING CONDITIONS:
0BA0  2191 ;               1. SHARE COUNT DECREASED FROM 1 TO 0, I.E. LAST PTE GONE
0BA0  2192 ;               2. READ OR WRITE COMPLETE
0BA0  2193 ;               3. PAGE CONTENTS DELETED (DELCONPFN)
0BA0  2194 ;               4. FAULTED OUT OF TRANSITION STATE
0BA0  2195 ;
0BA0  2196 ; CALLING SEQUENCE:
0BA0  2197 ;
0BA0  2198 ;       BSBW    MMG$DECPHDREF           ;R5 = PROCESS HEADER ADDRESS
0BA0  2199 ;       BSBW    MMG$DECPHDREF1          ;R1 = PROCESS HEADER VECTOR INDEX
0BA0  2200 ;
0BA0  2201 ; INPUT PARAMETERS:
0BA0  2202 ;
0BA0  2203 ;       DECPHDREF
0BA0  2204 ;               R5 = PROCESS HEADER ADDRESS FOR PAGE TABLE PAGE
0BA0  2205 ;
0BA0  2206 ;       DECPHDREF1
0BA0  2207 ;               R1 = PROCESS HEADER VECTOR INDEX
0BA0  2208 ;
0BA0  2209 ; IMPLICIT INPUTS:
0BA0  2210 ;
0BA0  2211 ;       NONE
0BA0  2212 ;
0BA0  2213 ; OUTPUT PARAMETERS:
0BA0  2214 ;
0BA0  2215 ;       DECPHDREF
0BA0  2216 ;               ONLY R1 ALTERED
0BA0  2217 ;
0BA0  2218 ;       DECPHDREF1
0BA0  2219 ;               ALL REGISTERS PRESERVED
0BA0  2220 ;
0BA0  2221 ; IMPLICIT OUTPUTS:
0BA0  2222 ;
0BA0  2223 ;       NONE
0BA0  2224 ;
0BA0  2225 ; COMPLETION CODES:
0BA0  2226 ;
0BA0  2227 ;       NONE
0BA0  2228 ;
0BA0  2229 ; SIDE EFFECTS:
0BA0  2230 ;
0BA0  2231 ;       NONE
0BA0  2232 ;
0BA0  2233 ;--
0BA0  2234
```

```
                     0BA0  2235 MMG$DECPHDREF::
                     0BA0  2236          MOVZWL  PHD$W_PHVINDEX(R5),R1   ;PROCESS HEADER VECTOR INDEX
                     0BA4  2237 MMG$DECPHDREF1::
                     0BA4  2238          DECW    @W^PHV$GL_REFCBAS[R1]   ;COUNT ONE LESS REFERENCE
                     0BA9  2239          BEQL    10$                     ;BRANCH IF THAT WAS THE LAST REFERENCE
                     0BAB  2240          RSB
                     0BAC  2241 10$:     BRW     SCH$SWPWAKE             ;INFORM THE SWAPPER, HEADER MAY GO
                     0BAF  2242                                          ;AND RETURN TO THIS ROUTINE'S CALLER
```

```
                        .SBTTL  INIBLDPKT - INIT FOR CALLING BUILDPKT
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE SETS UP R0-R2 FOR A SINGLE PAGE READ/WRITE
                ; TO THE ADDRESS SPECIFIED BY THE BACKING STORE ADDRESS.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW    MMG$INIBLDPKT
                ;
                ; INPUT PARAMETERS:
                ;
                ;       R2 = BACKING STORE ADDRESS
                ;       R3 = PAGE TABLE ENTRY ADDRESS (MASTER IF GLOBAL)
                ;       R5 = PROCESS HEADER ADDRESS (SYSTEM HEADER IF GLOBAL PAGE)
                ;               THIS IS ONLY USED FOR SECTION TYPE BACKING STORE ADDRESSES
                ;               IF THE BACKING STORE ADDRESS IN R2 IS KNOWN TO BE A
                ;               PAGING FILE ADDRESS, THEN IT SELF DESCRIBES AND THIS
                ;               PARAMETER IS IGNORED.
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       R0 = VIRTUAL BLOCK NUMBER
                ;       R1 = SECTION OR PAGE FILE CONTROL BLOCK ADDRESS
                ;       R2 = WINDOW ADDRESS
                ;       R3 = PAGE TABLE ENTRY ADDRESS (PRESERVED)
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       NONE
                ;
                ; COMPLETION CODES:
                ;
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;
                ;       NONE
                ;
                ;--
```

```
                                0BAF  2290 MMG$INIBLDPKT::
            21 52    17    E0  0BAF  2291          BBS     #PFN$V_GBLBAK,R2,10$     ;NOT AN I/O ADDRESS IF GBL BAK
            21 52    16    E0  0BB3  2292          BBS     #PTE$V_TYPO,R2,20$       ;BRANCH IF SECTION ADDRESS
      51    52    08    18  EF  0BB7  2293          EXTZV   #PFN$V_PGFLX,#PFN$S_PGFLX,R2,R1 ;PAGE FILE INDEX
         51  0000'DF41    D0  0BBC  2294          MOVL    @W^MMG$GL_PAGSWPVC[R1],R1 ;PAGE FILE CONTROL BLOCK ADDRESS
      50    52    16    00  EF  0BC2  2295          EXTZV   #PTE$V_PGFLVB,#PTE$S_PGFLVB,R2,R0 ;PAGE FILE VBN
            07 50    15    E1  0BC7  2296          BBC     #PTE$V_CHKPNT,R0,5$      ;SKIP NEXT CHECK IF BIT CLEAR
               7E    1C A1    D2  0BCB  2297          MCOML   PFL$L_MAXVBN(R1),-(SP)   ;SEE IF BIT IS PART OF LARGE VBN OR
                  50    8E    CA  0BCF  2298          BICL2   (SP)+,R0                 ; TURN IT OFF IF BIT REPRESENTS CHECKPOINTIN
                        24    12  0BD2  2299 5$:       BNEQ    40$                      ;BRANCH IF GOOD VBN
                                0BD4  2300 ;
                                0BD4  2301 ; INVALID BACKING STORE ADDRESS FOR I/O
                                0BD4  2302 ;
                                0BD4  2303 10$:      BUG_CHECK IVBAKADIO,FATAL        ;INVALID BACKING STORE ADR FOR I/O
                           FEFF 0BD4                 .WORD   ^XFEFF
                           0004' 0BD6                .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_IVBAKADIO!4
                                0BD8  2304 ;
                                0BD8  2305 ; SECTION TABLE BACKING STORE ADDRESS
                                0BD8  2306 ;
            52    52    32    0BD8  2307 20$:      CVTWL   R2,R2                    ;SECTION TABLE INDEX
      51    55    20 A5    C1  0BDB  2308          ADDL3   PHD$L_PSTBASOFF(R5),R5,R1 ;SECTION TABLE BASE ADDRESS
            51    6142    DE  0BE0  2309          MOVAL   (R1)[R2],R1              ;SECTION TABLE ENTRY ADDRESS
   50    53    00C8 C5    C3  0BE4  2310          SUBL3   PHD$L_POBR(R5),R3,R0     ;BYTE OFFSET FROM BASE OF PAGE TABLE
      50    50    FE 8F    78  0BEA  2311          ASHL    #-2,R0,R0                ;LONG WORD INDEX FROM PAGTBL BASE
52    08 A1    16    00    EF  0BEF  2312          EXTZV   #SEC$V_VPX,#SEC$S_VPX,SEC$L_VPXPFC(R1),R2 ;VIRTUAL PAGE NUMBER
            50    52    C2  0BF5  2313          SUBL    R2,R0                    ;RELATIVE PAGE IN SECTION
                              0BF8  2314 40$:
                              0BF8  2315          ASSUME  SEC$L_VBN EQ PFL$L_VBN
                              0BF8  2316          ASSUME  SEC$L_WINDOW EQ PFL$L_WINDOW
            50    10 A1    C0  0BF8  2317          ADDL    SEC$L_VBN(R1),R0         ;FORM FILE VBN
            52    0C A1    D0  0BFC  2318          MOVL    SEC$L_WINDOW(R1),R2      ;FILE WINDOW
                        05    0C00  2319          RSB
                              0C01  2320
                              0C01  2321          .END
```

```
...PFN                  = 00000860 R    03
ACVIOLAT                  000006C3 R    03
AST                       00000010
ASTPRM                    00000014
BADSYSPAG                 00000004 R    03
BAK                       00000024
BIT...                  = 00000003
BUG$_BADLCKWSLE           ******** X    03
BUG$_DECPTREF             ******** X    03
BUG$_DELWSLEX             ******** X    03
BUG$_FREWSLX              ******** X    03
BUG$_INCPTREF             ******** X    03
BUG$_IVBAKADIO            ******** X    03
BUG$_MAKEWSLE             ******** X    03
BUG$_MFYNULPGFL           ******** X    03
BUG$_PGFGBLBAD            ******** X    03
BUG$_PGFIPLHI             ******** X    03
BUG$_PGFLOCBAD            ******** X    03
BUG$_SCANDEADPT           ******** X    03
BUG$_WSLENOVAL            ******** X    03
BUG$_WSLVANVAL            ******** X    03
BUG$_WSSIZEERR            ******** X    03
CA$_MEASURE             = 00000002
CLUSTER                   00000020
CLU_END                   000004D9 R    03
CLU_END1                  000003EB R    03
CLU_END_RESRC             000004F3 R    03
CLU_END_RESRC1            000004D7 R    03
CLU_INI_INC               00000400 R    03
CLU_NXT                   0000040C R    03
CLU_SCRATCH_SIZ           0000004C
COUNT                     00000021
CTL$GL_IWSPEAK            ******** X    03
CTL$GL_PHD                ******** X    03
CTL$GL_WSPEAK             ******** X    03
DECVALWSLECNT             00000A32 R    03
DIOCNTWAIT_2              000001A0 R    03
DIR...                  = 00000001
DZROFPGWAIT_5             00000528 R    03
DZRO_GBL_SEC              00000532 R    03
DZRO_PROC_SEC             000001C7 R    03
DZRO_PTE                  0000052E R    03
DZRO_PTE_0                00000122 R    03
EXE$ACVIOLAT              ******** X    03
EXE$ALONONPAGED           ******** X    03
EXE$BUILDPKTR             ******** X    03
EXE$DEANONPAGED           ******** X    03
EXE$GL_FLAGS              ******** X    03
EXE$PAGRDERR              ******** X    03
EXE$V_NOCLUSTER           ******** X    03
EXE$V_TBCHK               ******** X    03
FLTCTL                    00000018
FLTPC                     00000020
FLTPSL                    00000024
FLTVA                     0000001C
FP_SAV                    0000003C
FREEPAGEWAIT              00000377 R    03
FREEPAGEWAIT1             00000398 R    03
FREPAGWAIT_5              000001B8 R    03
FRE_GBLTRANS              000007C4 R    03
GBLBAD                    00000150 R    03
GBLCRF                    0000018F R    03
GBLDZRO                   000001CA R    03
GBLDZRO_PGFL              00000146 R    03
GBLNOTRESIDENT            000001D9 R    03
GBLVALID                  00000130 R    03
GETPAGELOC                000000BA R    03
GET_IRP                   000001A5 R    03
GOT_IRP                   00000222 R    03
GPGTBL                    0000000F R    03
GPTX                      00000018
GPTX_PTE                  0000001C
INC1                      00000028
INC4                      0000002C
INC512                    00000030
INCPTREFBUG               00000B35 R    03
INCVALWSLECNT             00000ADB R    03
ININEWPFNWAIT             00000A4C R    03
IOC$GL_IRPFL              ******** X    03
IPL$_ASTDEL             = 00000002
IPL$_SYNCH              = 00000008
IPLHI                     00000000 R    03
IRP$B_PRI               = 00000023
IRP$C_LENGTH            = 000000C4
IRP$L_AST               = 00000010
IRP$L_ASTPRM            = 00000014
IRP$W_SIZE              = 00000008
IRPWAIT_3                 00000196 R    03
LCKWSLE_NOTPGTB           00000867 R    03
LOCBAD                    0000011E R    03
MMG$ALLOCPFN              ******** X    03
MMG$ALLOCSWPAREA          ******** X    03
MMG$AL_SYSPCB             ******** X    03
MMG$CRETVA                ******** X    03
MMG$DALCBAKSTORE          ******** X    03
MMG$DECPHDREF             00000BA0 RG   03
MMG$DECPHDREF1            00000BA4 RG   03
MMG$DECPTREF              00000B39 RG   03
MMG$DELWSLEPPG            00000A29 RG   03
MMG$DELWSLEX              00000A05 RG   03
MMG$FREWSLE               0000087F RG   03
MMG$FREWSLX               00000915 RG   03
MMG$FRE_TRYSKIP           00000942 RG   03
MMG$GB_FREWFLGS           ******** X    03
MMG$GL_GPTBASE            ******** X    03
MMG$GL_MAXGPTE            ******** X    03
MMG$GL_MAXPFN             ******** X    03
MMG$GL_PAGSWPVC           ******** X    03
MMG$GL_SPTBASE            ******** X    03
MMG$GL_SYSPHD             ******** X    03
MMG$INCPTREF              00000AFB RG   03
MMG$INIBLDPKT             00000BAF RG   03
MMG$ININEWPFN             00000A4F RG   03
MMG$LOCKPGTB              00000AF5 RG   03
```

```
MMG$MAKEWSLE              00000A91 RG     03      PFN$C_PPGTBL             = 00000004
MMG$PAGEFAULT             00000070 RG     03      PFN$C_PROCESS            = 00000000
MMG$PGFLTWAIT             000003B7 RG     03      PFN$C_RDINPROG           = 00000006
MMG$PGFLTWAIT_1           000003B2 R      03      PFN$C_RELPEND            = 00000003
MMG$PTEREF                ********     X  03      PFN$C_SYSTEM             = 00000001
MMG$REFCNTNEG             ********     X  03      PFN$M_BAK                = 007FFFFF
MMG$RELPFN                ********     X  03      PFN$M_DELCON             = 00000010
MMG$REMPFN                ********     X  03      PFN$M_GBLBAK             = 00800000
MMG$RESRCWAIT             0000039C RG     03      PFN$M_LOC                = 00000007
MMG$RLPFNSAVPTE           ********     X  03      PFN$M_MODIFY             = 00000080
MMG$SCNWSLX               ********     X  03      PFN$M_PAGTYP             = 00000007
MMG$SHRCNTNEG             ********     X  03      PFN$S_BAK                = 00000017
MMG$SVAPTECHK             ********     X  03      PFN$S_LOC                = 00000003
MMG$SVPCTX                0000036E RG     03      PFN$S_PAGTYP             = 00000003
MMG$V_NOWAIT              ********     X  03      PFN$S_PGFLX              = 00000008
MMG$WSLEPFN               000007B2 RG     03      PFN$V_BAK                = 00000000
MPH$INVALIDHK             0000093A RG     03      PFN$V_COLLISION          = 00000004
MPW$GL_WAITLIM            ********     X  03      PFN$V_DELCON             = 00000004
NOTGLOBAL                 000001D0 R      03      PFN$V_GBLBAK             = 00000017
NOTRANSITION              00000154 R      03      PFN$V_LOC                = 00000000
NOTSYSTEM                 00000024 R      03      PFN$V_MODIFY             = 00000007
NULLPGFL_NOMFY            00000959 R      03      PFN$V_PAGTYP             = 00000000
OP$_ACBL                = 000000F1                PFN$V_PGFLX              = 00000018
OP$_ACBW                = 0000003D                PFNLIST                    0000060C R     03
OP$_DECL                = 000000D7                PGF$V_LENVIO             = 00000000
OP$_DECW                = 000000B7                PGF$V_PGTBFLT            = 00000001
OP$_MOVL                = 000000D0                PGF$V_WRTACC             = 00000002
OP$_MOVW                = 000000B0                PGFCOMPLETE                00000606 R     03
OP$_MOVZWL              = 0000003C                PGFEXIT                    00000369 R     03
P0ADDR                    000000B4 R      03      PGFMONITOR                 0000003D R     03
PCB$B_PRIB              = 0000002F                PGFMONITOR1                000000A0 R     03
PCB$L_EFWM              = 0000004C                PHD$B_DFPFC              = 00000034
PCB$L_PHD               = 0000006C                PHD$B_PAGFIL             = 0000001F
PCB$L_WSSWP             = 00000020                PHD$B_PGTBPFC            = 00000035
PCB$W_DIOCNT            = 0000003E                PHD$L_P0BR               = 000000C8
PCB$W_GPGCNT            = 00000034                PHD$L_P1BR               = 000000D0
PCB$W_MTXCNT            = 0000000E                PHD$L_PAGEFLTS           = 0000004C
PCB$W_PPGCNT            = 00000036                PHD$L_PAGFIL             = 0000001C
PCB$W_STATE             = 0000002C                PHD$L_PGFLTIO            = 00000108
PCB_SAV                   00000040                PHD$L_PSTBASOFF          = 00000020
PFL$B_PFC               = 0000000B                PHD$L_PTWSLEVAL          = 00000068
PFL$L_MAXVBN            = 0000001C                PHD$V_IWSPEAKCK          = 00000004
PFL$L_VBN               = 00000010                PHD$V_NOACCVIO           = 00000003
PFL$L_WINDOW            = 0000000C                PHD$V_PFMFLG             = 00000000
PFM$MON                   ********     X  03      PHD$V_WSPEAKCHK          = 00000002
PFN$AB_STATE              ********     X  03      PHD$W_FLAGS              = 00000036
PFN$AB_TYPE               ********     X  03      PHD$W_PHVINDEX           = 00000042
PFN$AL_BAK                ********     X  03      PHD$W_PTCNTACT           = 00000070
PFN$AL_PTE                ********     X  03      PHD$W_PTCNTLCK           = 0000006C
PFN$AW_REFCNT             ********     X  03      PHD$W_PTCNTVAL           = 0000006E
PFN$AX_SHRCNT             ********     X  03      PHD$W_SWAPSIZE           = 00000052
PFN$AX_WSLX               ********     X  03      PHD$W_WSAUTH             = 0000000A
PFN$C_ACTIVE            = 00000007                PHD$W_WSDYN              = 0000000E
PFN$C_GBLWRT            = 00000003                PHD$W_WSFLUID            = 00000074
PFN$C_GLOBAL            = 00000002                PHD$W_WSLAST             = 00000012
PFN$C_GPGTBL            = 00000005                PHD$W_WSLIST             = 00000008
PFN$C_MFYPAGLST         = 00000001                PHD$W_WSNEXT             = 00000010
```

PHD$W_WSQUOTA	=	00000018			RELEASEPEND		0000060F	R		03
PHD$W_WSSIZE	=	00000050			RESOURCEWAIT		0000037E	R		03
PHV$C_REFCBAS		********	X	03	RSN$_ASTWAIT	=	00000001			
PHVREFCADR		00000048			RSN$_MPLEMPTY	=	0000000B			
PMS$AL_TRANSFLT		00000018	RG	02	RSN$_MPWBUSY	=	0000000C			
PMS$GL_DPTSCN		00000038	RG	02	RSN$_NPDYNMEM	=	00000003			
PMS$GL_DZROFLTS		00000014	RG	02	RSN$_SWPFILE	=	0000000A			
PMS$GL_FAULTS		00000000	RG	02	RSRCWAIT_3		00000125	R		03
PMS$GL_GVALID		0000003C	RG	02	SAVABS...	=	0000004C			
PMS$GL_PREADIO		00000008	RG	02	SCANDEADPT		000006D6	R		03
PMS$GL_PREADS		00000004	RG	02	SCH$GL_CURPCB		********		X	03
PMS$GL_PWRITES		0000000C	RG	02	SCH$GL_FREECNT		********		X	03
PMS$GL_PWRITIO		00000010	RG	02	SCH$GL_GROWLIM		********		X	03
PMS$GL_RDFLTS		00000004	RG	02	SCH$GL_MFYCNT		********		X	03
PPGTBL		00000063	R	03	SCH$GL_MFYLIM		********		X	03
PPGTBL_2		0000012B	R	03	SCH$GL_MFYLOLIM		********		X	03
PR$_IPL	=	00000012			SCH$GL_RESMASK		********		X	03
PR$_TBCHK	=	0000003F			SCH$GQ_COLPGWQ		********		X	03
PR$_TBIS	=	0000003A			SCH$GQ_FPGWQ		********		X	03
PRI		00000023			SCH$GQ_MWAIT		********		X	03
PROCPAG		00000362	R	03	SCH$GQ_PFWQ		********		X	03
PRT$C_EW	=	00000005			SCH$SWPWAKE		********		X	03
PSL$C_SUPER	=	00000002			SCH$WAITM		********		X	03
PSL$S_CURMOD	=	00000002			SEC$B_PFC	=	0000000B			
PSL$S_IPL	=	00000005			SEC$L_VBN	=	00000010			
PSL$V_CURMOD	=	00000018			SEC$L_VPXPFC	=	00000008			
PSL$V_IPL	=	00000010			SEC$L_WINDOW	=	0000000C			
PTE$C_KW	=	10000000			SEC$S_VPX	=	00000016			
PTE$M_DZRO	=	00020000			SEC$V_VPX	=	00000000			
PTE$M_GPTX	=	003FFFFF			SETSLAVEPTE		00000642	R		03
PTE$M_MODIFY	=	04000000			SGN$GL_BALSETCT		********		X	03
PTE$M_OWN	=	01800000			SGN$GL_PTPAGCNT		********		X	03
PTE$M_PFN	=	001FFFFF			SGN$GW_WSLMXSKP		********		X	03
PTE$M_PGFLVB	=	003FFFFF			SIZ...	=	00000001			
PTE$M_PROT	=	78000000			SIZE_TYPE		00000008			
PTE$M_TYP0	=	00400000			SS$_NORMAL	=	00000001			
PTE$M_TYP1	=	04000000			STATE		00000022			
PTE$M_VALID	=	80000000			SVAPTE		00000004			
PTE$S_GPTX	=	00000016			SWP$GL_BALBASE		********		X	03
PTE$S_OWN	=	00000002			SWP$GL_BSLOTSZ		********		X	03
PTE$S_PFN	=	00000015			SWP$GW_SWPINC		********		X	03
PTE$S_PGFLVB	=	00000016			SYSTEMSPACE		00000042	R		03
PTE$V_CHKPNT	=	00000015			TMP...	=	00000001			
PTE$V_CRF	=	00000010			TRANSITION		000000FF	R		03
PTE$V_DZRO	=	00000011			TRY_TO_CLUSTER		000003C3	R		03
PTE$V_GPTX	=	00000000			VA		0000000C			
PTE$V_MODIFY	=	0000001A			VA$M_BYTE	=	000001FF			
PTE$V_OWN	=	00000017			VA$S_VPN	=	00000015			
PTE$V_PFN	=	00000000			VA$V_P1	=	0000001E			
PTE$V_PGFLVB	=	00000000			VA$V_SYSTEM	=	0000001F			
PTE$V_TYP0	=	00000016			VA$V_VPN	=	00000009			
PTE$V_TYP1	=	0000001A			VALID		000000B1	R		03
PTE$V_VALID	=	0000001F			VBN		00000034			
PTEDAT		00000000			WINDOW		00000038			
QUEUE_PAGE_READ		0000033E	R	03	WQH$W_WQCNT	=	00000008			
READERR		0000065B	R	03	WQH$W_WQSTATE	=	0000000A			
READINPROG		00000382	R	03	WRITEINPROG		0000060F	R		03

```
WSL$C_GLOBAL                    = 00000004
WSL$C_GPGTBL                    = 0000000A
WSL$C_PPGTBL                    = 00000008
WSL$C_SYSTEM                    = 00000002
WSL$M_PAGTYP                    = 0000000E
WSL$M_VALID                     = 00000001
WSL$M_WSLOCK                    = 00000020
WSL$S_PAGTYP                    = 00000003
WSL$V_PAGTYP                    = 00000001
WSL$V_VALID                     = 00000000
WSL$V_WSLOCK                    = 00000005
WSLENOVAL                         0000086F R      03
WSLEPFNMSK                        000007AE R      03
WSLVANVAL                         000007DA R      03
WSSIZEERR                         0000086B R      03
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	0000004C (76.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$210	00000040 (64.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
MMGCOD	00000C01 (3073.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
Z$INIT$PFN_FIXUP_TABLE	00000036 (54.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.05	00:00:02.27
Command processing	121	00:00:00.51	00:00:07.06
Pass 1	571	00:00:21.08	00:01:14.69
Symbol table sort	0	00:00:02.88	00:00:08.42
Pass 2	414	00:00:06.48	00:00:21.46
Symbol table output	1	00:00:00.24	00:00:00.83
Psect synopsis output	0	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1140	00:00:31.26	00:01:54.75

```
The working set limit was 2100 pages.
152632 bytes (299 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1753 non-local and 129 local symbols.
2321 source lines were read in Pass 1, producing 32 object records in Pass 2.
39 pages of virtual memory were used to define 36 macros.
```

```
                                    +----------------------------+
                                    ! Macro library statistics !
                                    +----------------------------+

Macro library name                                  Macros defined
------------------                                  --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                              20
_$255$DUA28:[SYSLIB]STARLET.MLB;2                           13
TOTALS (all libraries)                                      33

1851 GETS were required to define 33 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:PAGEFAULT/OBJ=OBJ$:PAGEFAULT MSRC$:PAGEFAULT/UPDATE=(ENH$:PAGEFAULT)+EXECML$/LIB
```

0378 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PAGEFAULT
LIS

OSWPSCHED
LIS

PARAMETER
LIS

P1SYSVECT
LIS

PAGEFILE
LIS